DTIC FILE COPY

(2)

NAVAL POSTGRADUATE SCHOOL

Monterey, California

AD-A205 308

DTIC
ELECTE
MAR 1 0 1989
S D
H

THESIS

ARMA MODELING

by

Gurhan Kayahan

December 1988

Thesis Advisor Ralph Hippenstiel

Approved for public release; distribution is unlimited.

89 3 09 031

Unclassified

security classification of this page

REPORT DOCUMENTATION PAGE

1a Report Security Classification: Unclassified			1b Restrictive Markings
2a Security Classification Authority			3 Distribution Availability of Report: Approved for public release; distribution is unlimited.
2b Declassification Downgrading Schedule			
4 Performing Organization Report Number(s)			5 Monitoring Organization Report Number(s)
6a Name of Performing Organization: Naval Postgraduate School	6b Office Symbol (*if applicable*): 32		7a Name of Monitoring Organization: Naval Postgraduate School
6c Address (*city, state, and ZIP code*): Monterey, CA 93943-5000			7b Address (*city, state, and ZIP code*): Monterey, CA 93943-5000
8a Name of Funding Sponsoring Organization	8b Office Symbol (*if applicable*)		9 Procurement Instrument Identification Number
8c Address (*city, state, and ZIP code*)			10 Source of Funding Numbers: Program Element No / Project No / Task No / Work Unit Accession No

11 Title (*Include security classification*): ARMA MODELING

12 Personal Author(s): Gurhan Kayahan

13a Type of Report	13b Time Covered	14 Date of Report (*year, month, day*)	15 Page Count
Master's Thesis	From To	December 1988	89

16 Supplementary Notation: The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17 Cosati Codes			18 Subject Terms (*continue on reverse if necessary and identify by block number*)
Field	Group	Subgroup	ARMA Modeling, Yule-Walker Equations, Cholesky Decomposition

19 Abstract (*continue on reverse if necessary and identify by block number*)

This thesis estimates the frequency response of a network where the only data is the output obtained from an Autoregressive-moving average (ARMA) model driven by a random input.

Models of random processes and existing methods for solving ARMA models are examined. The estimation is performed iteratively by using the *Yule-Walker Equations* in three different methods for the AR part and the *Cholesky factorization* for the MA part. The AR parameters are estimated initially, then MA parameters are estimated assuming that the AR parameters have been compensated for. After the estimation of each parameter set, the original time series is filtered via the inverse of the last estimate of the transfer function of an AR model or MA model, allowing better and better estimation of each model's coefficients. The iteration refers to the procedure of removing the MA or AR part from the random process in an alternating fashion allowing the creation of an almost pure AR or MA process, respectively. As the iteration continues the estimates are improving. When the iteration reaches a point where the coefficients converge the last MA and AR model coefficients are retained as final estimates. (KR) ←

20 Distribution Availability of Abstract	21 Abstract Security Classification	
☒ unclassified unlimited ☐ same as report ☐ DTIC users	Unclassified	
22a Name of Responsible Individual: Ralph Hippenstiel	22b Telephone (*include Area code*): (408) 646-2768	22c Office Symbol: 62Hi

DD FORM 1473,84 MAR 83 APR edition may be used until exhausted
All other editions are obsolete

security classification of this page

Unclassified

Approved for public release; distribution is unlimited.

ARMA Modeling

by

Gurhan Kayahan
Lieutenant Junior Grade,Turkish Navy
B.S., Turkish Naval Academy, 1982

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1988

Author: _____
Gurhan Kayahan

Approved by: _____
Ralph Hippenstiel, Thesis Advisor

_____
Murali Tummala, Second Reader

_____
John P. Powers, Chairman,
Department of Electrical and Computer Engineering

_____
Gordon E. Schacher,
Dean of Science and Engineering

ABSTRACT

This thesis estimates the frequency response of a network where the only data is the output obtained from an Autoregressive-moving average (ARMA) model driven by a random input.

Models of random processes and existing methods for solving ARMA models are examined. The estimation is performed iteratively by using the *Yule-Walker Equations* in three different methods for the AR part and the *Cholesky factorization* for the MA part. The AR parameters are estimated initially, then MA parameters are estimated assuming that the AR parameters have been compensated for. After the estimation of each parameter set, the original time series is filtered via the inverse of the last estimate of the transfer function of an AR model or MA model, allowing better and better estimation of each model's coefficients. The iteration refers to the procedure of removing the MA or AR part from the random process in an alternating fashion allowing the creation of an almost pure AR or MA process, respectively. As the iteration continues the estimates are improving. When the iteration reaches a point where the coefficients converge the last MA and AR model coefficients are retained as final estimates.

Accession For
NTIS GRA&I ☑
DTIC TAB ☐
Unannounced ☐
Justification
By
Distribution/
Availability Codes
Dist Special
A-1

QUALITY INSPECTED 4

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

A significant debt of gratitude is owed to Professor Ralph Hippenstiel, for the many hours of assistance and guidance he has extented, in the preparation of this thesis.

Also I would like to express my sincere appreciation to Professor Murali Tummala of the Department of Electrical and Computer Engineering of the Naval Postgraduate School, my second reader.

To my wife, Ayse Kayahan and my son, Ecmel Kayahan, for their encouragement and patience, I am deeply grateful.

Finally, I wish to express my appreciation to the Turkish Navy Authority for the opportunity to study in the Naval Postgraduate School.

I. INTRODUCTION

In this thesis, an iterative approach is presented to estimate the frequency response of a network where the only data is the output obtained from an Autoregressive-moving average (ARMA) model driven by a random input. The AR parameters are estimated initially, then the MA parameters are estimated assuming that the AR parameters have been compensated for..

To find the frequency response of an ARMA network two problems have to be addressed. One is due to the shortness of the observed data (time limitation) and the resulting distortion of the estimated correlation function. The second problem is due to the nonlinear combination of the coefficients of the MA part as they appear in the estimate of the correlation function.

To achieve this, we estimate coefficients of MA part and AR part iteratively. We use the Yule-Walker Method for AR coefficients estimation and the Cholesky decomposition for MA part estimation. We assume that the ARMA network output, i.e., the observed signal is produced by white Gaussian noise driving the network. Three different methods are used to estimate the AR coefficients. To minimize the effect of the MA part in the correlation function, correlation lags greater than the correlation length of the MA part are used in estimating the initial AR coefficients. The iteration refers to the procedure, which removes estimated MA or estimated AR contribution from the output of the ARMA model in an alternating fashion. This allows better and better estimates of the AR and MA coefficients as the iteration continues. For the iterative AR coefficient estimation three methods can be used, denoted by method 1, method 2 and method 3. Method 1, 2 and 3 use correlation lags starting at $p+1$, 0, and 0 respectively, where the first two methods use a square matrix inverse and the third method uses a Pseudo matrix inverse.

Introduction to models of random processes is presented in Chapter II. Existing techniques to solve for the parameters of ARMA models are presented in Chapter II. Chapter IV includes simulation results. Simulation studies employed the Matlab package on the IBM PC/AT. Conclusions and recommendations are presented in Chapter V.

II. MODELS OF RANDOM PROCESSES

A. AR PROCESS MODEL

We say that x(n) is an autoregressive process of order p, or simply an AR(p) process if it satisfies the difference equation,

$$x(n) + a_1 x(n-1) + ... + a_p x(n-p) = \varepsilon(n) \tag{2.1}$$

where $a_1, a_2, ..., a_p$ are constant coefficients, and $\varepsilon(n)$ is a pure random process [Ref. 1].

Equation (2.1) may be written in the following form,

$$x(n) = -\sum_{k=1}^{p} a_k x(n-k) + \varepsilon(n) \tag{2.2}$$

A realization of Eq.(2.2) is illustrated in Figure 1.

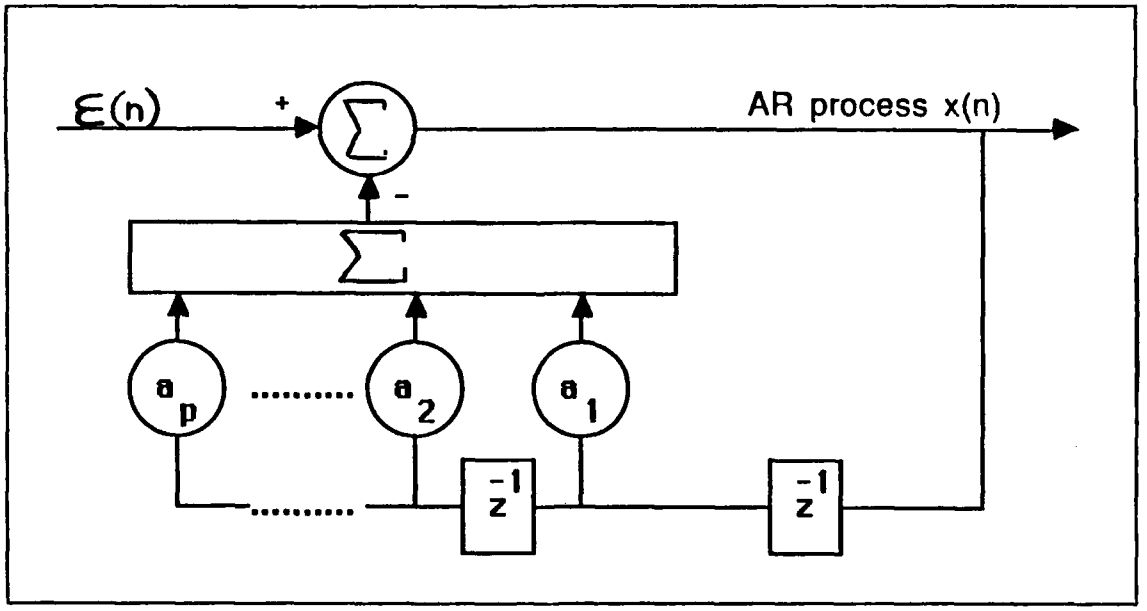

Figure 1. Autoregressive Model of Order p

The system transfer function H(z) between the input $\varepsilon(n)$ and the output x(n) for the AR model shown in Figure 1 is

$$H(z) = \frac{1}{A(z)} = \frac{1}{1 + a_1 z^{-1} + a_2 z^{-2} + ... + a_p z^{-p}} \tag{2.3}$$

It is required that A(z) has all its roots within the unit circle in the z-plane which guarantees that H(z) is a stable and causal filter.

The form of equation (2.3) illustrates that AR models have finite poles but no zeroes. Hence, this model is sometimes called an *All-Pole* model. Because an AR model has no poles outside or on the unit circle, it has the strict minimum-delay property and hence is always invertible. In general, minimum delay means that the transfer function must have no poles outside the unit circle, but can have poles on the unit circle. Strictly minimum delay means that the transfer function has no poles outside or on the unit circle.

The AR model is also called an *Infinite Impulse Response* (IIR) filter. According to definition (2.2), output x(n) depends on past values of the output and on the present input. Because of this, it is also referred to as a *Pure Feedback* system.

The power spectral density for AR models is given by

$$P_{AR}(f) = \frac{\sigma_\varepsilon^2}{|A(f)|^2} \tag{2.4}$$

where

$$A(f) = 1 + \sum_{k=1}^{p} a_k e^{-j\pi fkT}$$

and T is the sampling interval [Ref. 2].

B. MA PROCESS MODEL

The sequence x(n) is said to be a moving average process of order q (denoted by MA(q)) if it satisfies the difference equation,

$$x(n) = b_0\varepsilon(n) + b_1\varepsilon(n-1) + ... + b_q\varepsilon(n-q) \tag{2.5}$$

where $b_0, b_1, ..., b_q$ are coefficients, and $\varepsilon(n)$ is a pure random process [Ref. 1].

Equivalently, we may write,

$$x(n) = \sum_{k=0}^{q} b_k \varepsilon(n-k) \tag{2.6}$$

We may say that the output of an MA model depends only on present and past values of the input, i.e., there is no feedback in an MA model.

A realization of Eq. (2.6) is illustrated in Figure 2.

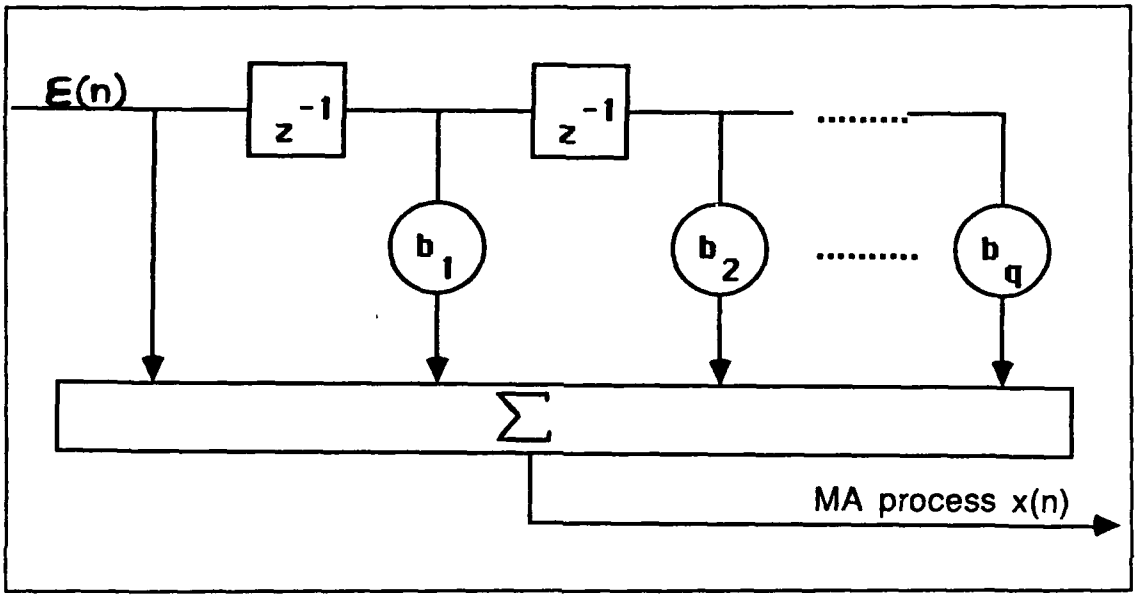

Figure 2. Moving Average Model of Order q

The system transfer function H(z) between the input $\varepsilon(n)$ and the output x(n) for the MA process is

$$H(z) = B(z) = b_0 + b_1 z^{-1} + b_2 z^{-2} + ... + b_q z^{-q} \tag{2.7}$$

This transfer function has q finite zeroes, but no poles. Hence, the MA model is also called an *All-Zero* model. MA models are invertible if and only if B(z) has no zeroes outside the unit circle, nor on the unit circle.

MA models are also called *Finite Impulse Response* (FR) filters.

The power spectral density for an MA model is given by

$$P_{MA}(f) = \sigma_\varepsilon^2 |B(f)|^2 \tag{2.8}$$

where

$$B(f) = b_0 + \sum_{k=1}^{q} b_k e^{-j\pi fkT}$$

and T is the sampling interval.

C. ARMA PROCESS MODEL

We say that x(n) is an autoregressive-moving average process of order (p,q) or simply an ARMA(p,q) process if it is satisfies the difference equation,

$$x(n) + a_1 x(n-1) + ... + a_p x(n-p) = b_0 \varepsilon(n) + b_1 \varepsilon(n-1) + ... + b_q \varepsilon(n-p) \quad (2.9)$$

where, again, $a_1, ..., a_p, b_0, ..., b_q$ are coefficients and $\varepsilon(n)$ is a pure random process [Ref. 1].

Equation (2.9) may be written as,

$$x(n) = -\sum_{k=1}^{p} a_k x(n-k) + \sum_{k=0}^{q} b_k \varepsilon(n-k) = \sum_{k=0}^{\infty} h_k \varepsilon(n-k) \quad (2.10)$$

The assumption $b_0 = 1$ can be made without any loss of generality because the input $\varepsilon(n)$ can always be scaled to account for any filter gain [Ref. 2].

A realization of Eq. (2.10) is illustrated in Figure 3.

The system transfer function for the ARMA process is given by

$$H(z) = \frac{B(z)}{A(z)} = \frac{1 + b_1 z^{-1} + b_2 z^{-2} + ... + b_q z^{-q}}{1 + a_1 z^{-1} + a_2 z^{-2} + ... + a_p z^{-p}} \quad (2.11)$$

where A(z) is the z-transform of the AR part and B(z) is the z-transform of the MA part.

Both polynomials A(z) and B(z) are assumed to have all of their zeros within the unit circle of the z-plane to guarantee that H(z) is a stable minimum-phase invertible filter.

The power spectral density for ARMA models is given by [Ref. 2]

$$P_{ARMA}(f) = \sigma_\varepsilon^2 \left| \frac{B(f)}{A(f)} \right|^2 \quad (2.12)$$

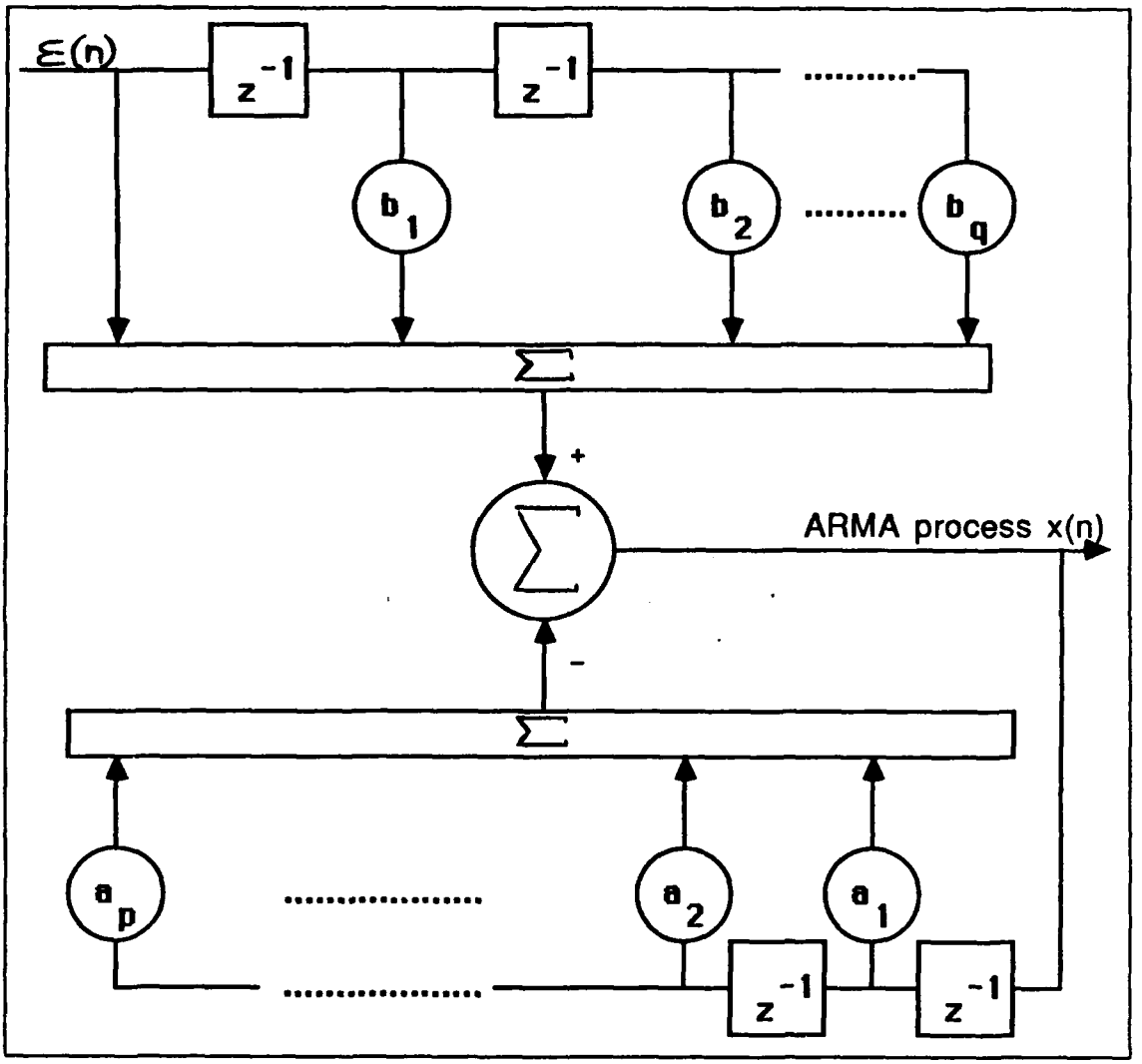

Figure 3. Autoregressive-Moving Average Model of Order (p,q)

D. RELATIONSHIPS OF RANDOM PROCESSES

The Wold decomposition theorem [Ref. 3] relates the AR, MA and ARMA models. It shows that, if the Power Spectral Density is purely continuous, any AR or ARMA process can be represented by a unique MA model of infinite order.

Another important theorem which is stated by Kolmogorov [Ref. 4] says that any ARMA or MA process can be represented by an AR process of infinite order.

To illustrate these theorems, we model [Ref. 5] an ARMA(1,1) process by an $AR(\infty)$ or by an $MA(\infty)$ process. From equation (2.11), the system transfer function for the ARMA(1,1) process is

$$H(z) = \frac{1 + b_1 z^{-1}}{1 + a_1 z^{-1}}$$

If we use $AR(\infty)$ process to represent ARMA(1,1) process, where

$$H(z) = \frac{1}{1 + c_1 z^{-1} + c_2 z^{-2} + \ldots}$$

and

$$C(z) = 1 + \sum_{k=1}^{\infty} c_k z^{-k} = \frac{1 + a_1 z^{-1}}{1 + b_1 z^{-1}}$$

By using synthetic division we find that

$$C(z) = 1 + (a_1 - b_1)z^{-1} + (b_1^2 - a_1 b_1)z^{-2} + (b_1^3 - a_1 b_1^2)z^{-3} + \ldots$$

Hence inverse z-transform of az^{-m} is $a\delta(k - m)$ [Ref. 6] and

$$\delta(k) = \begin{cases} 1 & \text{if } k = 0 \\ 0 & \text{else,} \end{cases}$$

the inverse z-transform of C(z) is

$$c_k = \delta(k) + (a_1 - b_1)\delta(k - 1) + (b_1^2 - a_1 b_1)\delta(k - 2) + (b_1^3 - a_1 b_1^2)\delta(k - 3) + \ldots$$

or

$$c_k = \begin{cases} 1 & \text{if } k = 0 \\ (a_1 - b_1)(-b_1)^{k-1} & \text{if } k \geq 1 \end{cases}$$

If we use a finite order AR(p) we should choose p to satisfy $c_{p+1} \approx 0$ or, equivalently $b_1^p \approx 0$.

Therefore, a high-order AR model will be required when the zero of the ARMA process gets closer to the unit circle.

In a similar way if we use an $MA(\infty)$ process to represent an ARMA(1,1) process, let

$$H(z) = d_0 + d_1 z^{-1} + d_2 z^{-2} + \ldots$$

where

$$D(z) = \sum_{k=0}^{\infty} d_k z^{-k} = \frac{1 + b_1 z^{-1}}{1 + a_1 z^{-1}}$$

By using synthetic division as we did above, the inverse z-transform of D(z) will be

$$d_k = \begin{cases} 1 & \text{if } k = 0 \\ (b_1 - a_1)(-a_1)^{k-1} & \text{if } k \geq 1 \end{cases}$$

If we use a finite order MA(q) model, we should choose q to satisfy $d_{q-1} \approx 0$ or, equivalently $a_1^q \approx 0$.

Therefore, a high-order MA model will be required when the pole of the ARMA process gets closer to the unit circle.

E. RELATIONSHIP OF AR, MA, AND ARMA PARAMETERS TO THE AUTOCORRELATION SEQUENCE

In this section, we will present the relationship of the model parameters to the autocorrelation sequence [Ref. 2].

If we multiply Eq. (2.10) by $x^*(n-m)$ and take expectation, the result will be

$$E\{x(n)x^*(n-m)\} = -\sum_{k=1}^{p} a_k E\{x(n-k)x^*(n-m)\} + \sum_{k=0}^{q} b_k E\{\varepsilon(n-k)x^*(n-m)\} \quad (2.13)$$

where the superscript * is used to denote the complex conjugation.

Equation (2.13) may be written as,

$$r_{xx}(m) = -\sum_{k=1}^{p} a_k r_{xx}(m-k) + \sum_{k=0}^{q} b_k r_{\varepsilon x}(m-k) \quad (2.14)$$

The cross correlation between the input and the output can be written as,

$$r_{\varepsilon x}(l) = E\{\varepsilon(n+l)x^*(n)\} = E\left\{\varepsilon(n+l)\left[\varepsilon^*(n) + \sum_{k=1}^{\infty} h_k^* \varepsilon^*(n-k)\right]\right\}$$

$$r_{\varepsilon x}(l) = r_{\varepsilon\varepsilon}(l) + \sum_{k=1}^{\infty} h_k^* r_{\varepsilon\varepsilon}(l+k) \tag{2.15}$$

where $h_0 = 1$ by definition (Eq. 2.10).

If we assume that the driving sequence is a white noise process of zero mean and variance σ_ε^2 then [Ref. 2]

$$r_{\varepsilon x}(l) = \begin{cases} 0 & \text{for } l > 0 \\ \sigma_\varepsilon^2 & \text{for } l = 0 \\ \sigma_\varepsilon^2 h^*_{(-l)} & \text{for } l < 0 \end{cases} \tag{2.16}$$

When we substitute Eq. (2.16) in Eq. (2.14), we get final relationship between the ARMA parameters and the autocorrelation sequence.

$$r_{xx}(m) = \begin{cases} r_{xx}^*(-m) & \text{for } m < 0 \\ -\sum_{k=1}^{p} a_k r_{xx}(m-k) + \sigma_\varepsilon^2 \sum_{k=m}^{q} b_k h^*_{(k-m)} & \text{for } 0 \le m \le q \\ -\sum_{k=1}^{p} a_k r_{xx}(m-k) & \text{for } m > q \end{cases} \tag{2.17}$$

The relationship between the autocorrelation sequence and a pure autoregressive model may be written by setting $q = 0$ in Eq. (2.17)

$$r_{xx}(m) = \begin{cases} -\sum_{k=1}^{p} a_k r_{xx}(m-k) & \text{for } m > 0 \\ -\sum_{k=1}^{p} a_k r_{xx}(-k) + \sigma_\varepsilon^2 & \text{for } m = 0 \\ r_{xx}^*(-m) & \text{for } m < 0 \end{cases} \tag{2.18}$$

The relationship between the autocorrelation sequence and a pure moving average model may be written by setting $p = 0$ in Eq. (2.17)

$$r_{xx}(m) = \begin{cases} 0 & \text{for} \quad m > q \\ \sigma_\epsilon^2 \sum_{k=m}^{q} b_k b^*_{(k-m)} & \text{for} \quad 0 \le m \le q \\ r_{xx}^*(-m) & \text{for} \quad m < 0 \end{cases} \tag{2.19}$$

In this case we should note that

$$h_k = b_k \quad \text{for} \quad 1 \le k \le q$$

III. TECHNIQUES TO SOLVE FOR THE PARAMETERS OF ARMA MODELS

The estimation of the parameters of ARMA processes is a classical problem which is still being investigated by statisticians. Methods to find the parameters for purely AR processes are well known, but for ARMA processes some problems remain.

There is a nonlinear relationship between the ARMA parameters and the autocorrelation of process x(n). The nonlinear equation (2.17) presents the difficulty of estimating the ARMA parameters, even when we know the autocorrelation sequence exactly. Techniques based on iterative maximum likelihood estimation (MLE) can be used to find the ARMA parameters. These techniques require complex computations and are not guaranteed to converge, or they may converge to the wrong solution. Therefore they are not practical for real time series. For AR parameters, techniques based on the least squares criterion lead to solutions of linear equations and hence reduce the computational complexity. Unfortunately, the moving average parameters of an ARMA model cannot be found easily by solving a set of linear equations. The MA parameters are convolved with the impulse response coefficients h(k) which causes a nonlinear relationship between the autocorrelation sequence and the filter coefficients.

In section III-A and III-B, we will discuss the methods of AR and MA parameter estimation, and in section III-C we will present an iterative approach to find the parameters.

A. AR PARAMETER ESTIMATION

In this section, we present three widely used methods of extracting the model parameters from a given block of measured data x(n).

These methods are:

1. The autocorrelation, or Yule-Walker method
2. The covariance method
3. Burg's method

All three methods of estimating AR parameters are based on least-squares minimization criteria obtained by replacing the ensemble averages by appropriate time averages [Ref. 7].

The criteria for the optimal forward ($e_p^+(n)$) and backward ($e_p^-(n)$) predictors are obtained by minimizing

$$E\left[e_p^+(n)^2\right] \qquad \text{and} \qquad E\left[e_p^-(n)^2\right]$$

where $e_p^+(n)$ and $e_p^-(n)$ are the result of filtering x_n through the prediction-error filter as given by

$$e_p^+(n) = x_n + a_{p_1}x_{n-1} + a_{p_2}x_{n-2} + \ldots\ldots\ldots + a_{pp}x_{n-p}$$

$$e_p^-(n) = x_{n-p} + a_{p_1}x_{n-p+1} + a_{p_2}x_{n-p+2} + \ldots\ldots\ldots + a_{pp}x_n$$

The autocorrelation method is the most obvious and straightforward one. Equation (2.18) may be evaluated for the p+1 lag indices $0 \le m \le p$ and put into the following matrix form

$$\begin{bmatrix} r_{xx}(0) & r_{xx}(-1) & r_{xx}(-2) & r_{xx}(-p) \\ r_{xx}(1) & r_{xx}(0) & r_{xx}(-1) & r_{xx}(-p+1) \\ r_{xx}(2) & r_{xx}(1) & r_{xx}(0) & r_{xx}(-p+2) \\ \vdots & \vdots & \vdots & \vdots \\ r_{xx}(p) & r_{xx}(p-1) & r_{xx}(p-2) & r_{xx}(0) \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} \sigma_\varepsilon^2 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{3.1}$$

By noting that

$$r_{xx}(-k) = r_{xx}^*(k)$$

Eq. (3.1) can be written as

$$\begin{bmatrix} r_{xx}(0) & r_{xx}(1) & r_{xx}(2) & r_{xx}(p) \\ r_{xx}(1) & r_{xx}(0) & r_{xx}(1) & r_{xx}(p-1) \\ r_{xx}(2) & r_{xx}(1) & r_{xx}(0) & r_{xx}(p-2) \\ \vdots & \vdots & \vdots & \vdots \\ r_{xx}(p) & r_{xx}(p-1) & r_{xx}(p-2) & r_{xx}(0) \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} \sigma_\varepsilon^2 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{3.2}$$

From Eq. (3.2)

$$\sigma_\varepsilon^2 = \sum_{i=0}^{p} r_{xx}(i)a_i \quad \textit{where} \quad a_0 = 1$$

We can replace the autocorrelations $r_{xx}(k)$ by the corresponding sample autocorrelations (biased estimate) computed from the given block of data, where

$$\hat{r}_{xx}(k) = \frac{1}{N} \sum_{n=0}^{N-1-k} x_{n+k}^{*} x_n \qquad \text{for} \quad 0 \le k \le p \tag{3.3}$$

We use the biased autocorrelation estimator because the unbiased autocorrelation estimates may result in autocorrelation matrices that are not positive semidefinite, which means certain matrix equations have no solution. On the other hand, the autocorrelation matrices formed from the biased autocorrelation estimate will always be positive semidefinite [Ref. 2].

By solving the normal equations, we can obtain estimates of the model's parameters $\{a_1, a_2, a_3, \ldots, a_p, \sigma_t^2\}$. Equation (3.1) is known as the *AR, Yule-Walker or Normal Equations* . The autocorrelation matrix of this equation is both Toeplitz and Hermitian because

$$r_{xx}(-k) = r_{xx}^{*}(k)$$

The solution of the Hermitian Toeplitz equations can be computed with the *Levinson Algorithm*. This algorithm is a lattice realization for linear prediction filters. We illustrate the prediction-error sequence in Figure 4.

The p^{th} prediction error is given by

$$e_p(n) = x_n + a_{p_1} x_{n-1} + a_{p_2} x_{n-2} + \ldots\ldots + a_{pp} x_{n-p} \tag{3.4}$$

Looking at the first two prediction errors

$$e_1(n) = x_n + a_{11} x_{n-1}$$

$$e_2(n) = x_n + a_{21} x_{n-1} + a_{22} x_{n-2}$$

where $(1, a_{11})$, and $(1, a_{21}, a_{22})$ represent the best predictors of orders p = 1 and p = 2, respectively. The extra index is used to indicate the order of the predictor.

In the autocorrelation method the ensemble average in minimization of the forward prediction error is replaced by the least-squares time average criteria $\hat{\varphi}$ as follows

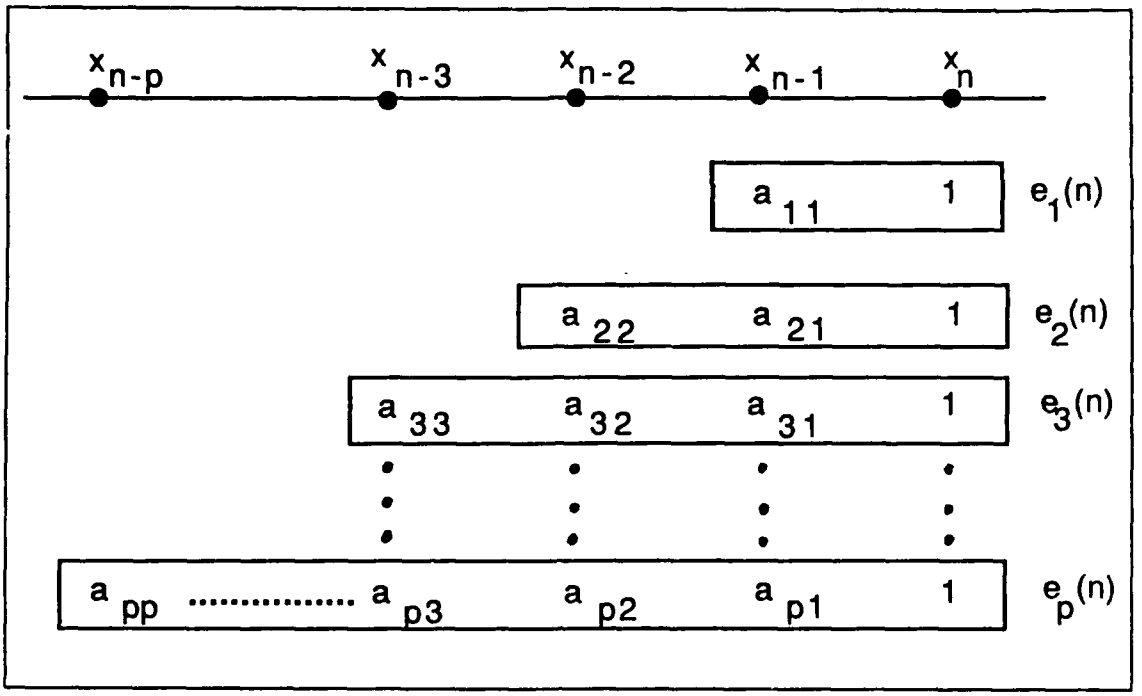

Figure 4. Prediction-error Sequence

$$\hat{\varphi} = \frac{1}{N} \sum_{n=0}^{N+p-1} |e_p^+(n)|^2$$
$$= \frac{1}{N} \sum_{n=0}^{N+p-1} \left| x_n + \sum_{k=1}^{p} a_{pk} x_{n-k} \right|^2 \qquad (3.5)$$

where it is assumed that the data $x_0, x_1, \ldots\ldots, x_{N-1}$ are observed and $x_n = 0$ for outside the range $0 \le n \le N-1$. The minimization of the time-average criteria with respect to the real and imaginary part of the a_{pk} 's will lead exactly to the same set of Yule-Walker Equations (3.1). One way to solve these equations is via the Levinson recursion which is an iterative technique that extracts the next order predictor from the previous one.

The Levinson algorithm can be summarized as follows [Ref. 7]:

1- Initialize the recursion at p = 0, by setting

$$A_0(z) = 1 \qquad \text{and} \qquad E_0 = r_{xx}(0) = E[x_n^2]$$

where $A_0(z)$ is the prediction-error filter and E_0 is the mean-squared prediction error at the zero stage. So, initially we have no prediction. At stage p,

the prediction-error filter will be $A_p(z)$ and the mean-squared prediction error will be E_p.

2- Compute Υ_{p+1} which is called the *reflection* or PARCOR coefficient:

$$\Upsilon_{p+1} = \frac{\sum_{l=0}^{p} a_{pl} r(p+1-l)}{\sum_{l=0}^{p} a_{pl} r(l)} \tag{3.6}$$

3- Recursively determine the $(p+1)^{th}$ order prediction-error filter polynomial $A_{p+1}(z)$:

$$A_{p+1}(z) = A_p(z) - \Upsilon_{p+1} z^{-(p+1)} A_p(z^{-1}) \tag{3.7}$$

4- Update the mean-squared prediction error:

$$E_{p+1} = (1 - \Upsilon_{p+1}^2) E_p \tag{3.8}$$

5- Continue the iteration until the final desired order is reached.

If the process x(n) is AR(p), then iteration will continue up to order p. It will provide the AR coefficients $a_{1p}, a_{2p}, \ldots\ldots, a_{pp}$ which are also the best prediction coefficients. If we continue iteration after p, all prediction coefficients of order higher than p will be close to zero [Ref. 7].

Although the autocorrelation method is the most obvious and efficient one, and the resulting prediction-error filter is guaranteed to be minimal phase, it suffers from the effect of prewindowing the data sequence x(n) by padding it with zeros to the left and to the right. This reduces the accuracy of the method, especially when we have short data records.

In the Covariance method the ensemble average in minimization of the forward prediction error is replaced by the time average as follows

$$\hat{\varphi} = \frac{1}{N-p} \sum_{n=p}^{N-1} \left| x_n + \sum_{k=1}^{p} a_{pk} x_{n-k} \right|^2 \tag{3.9}$$

The only difference between this method and autocorrelation method is in the limits of the summation. In the covariance method we observe all the data points needed to compute $\hat{\varphi}$. Since we do not need the data outside the range $0 \le n \le N-1$ we do not

need to set x_n explicitely to zero. For that reason the covariance method seems to be more realistic.

If we perform the minimization, we find the AR parameter estimates as the solution of equations (3.10) [Ref. 5].

$$\begin{bmatrix} c_{xx}(1,1) & c_{xx}(1,2) & & c_{xx}(1,p) \\ c_{xx}(2,1) & c_{xx}(2,2) & & c_{xx}(2,p) \\ \vdots & \vdots & & \vdots \\ c_{xx}(p,1) & c_{xx}(p,2) & & c_{xx}(p,p) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = - \begin{bmatrix} c_{xx}(1,0) \\ c_{xx}(2,0) \\ \vdots \\ c_{xx}(p,0) \end{bmatrix} \tag{3.10}$$

where

$$c_{xx}(j,k) = \frac{1}{N-p} \sum_{n=p}^{N-1} x_{n-j}^{*} x_{n-k}$$

Note that $c_{xx}(j,k)$ is an estimate of $r_{xx}(j-k)$. The $c_{xx}(j,k)$ uses the sum of only N-p lag products to estimate the autocorrelation function for each lag even more lags are available. In contrast in the estimation of $r_{xx}(0)$ the autocorrelation method uses all data point, while the covariance method uses only N-p data points in the summation. The minimization gives us a non-Toeplitz $c_{xx}(j,k)$ matrix. This implies that we can not use the Levinson algorithm to solve Eq. (3.10). The equations may be solved by using the Cholesky decomposition which will be computationally more expensive. The estimated poles using this method are not guaranteed to lie within the unit circle. As an example, consider a first order predictor (p = 1) and a length-three (N = 3) sequence [Ref. 7]. The time-average criteria will be

$$\hat{\varphi} = \frac{1}{2} \sum_{n=1}^{2} \left| e_1^{+}(n) \right|^2 = \frac{1}{2} \left((x_1 + a_{11} x_0)^2 + (x_2 + a_{11} x_1)^2 \right)$$

When we minimize $\hat{\varphi}$ by differentiating it with respect to a_{11} and setting the derivative to zero , we have

$$(x_1 + a_{11} x_0) x_0 + (x_2 + a_{11} x_1) x_1 = 0$$

$$a_{11} = - \frac{x_1 x_0 + x_2 x_1}{x_0^2 + x_1^2}$$

Note the denominator does not depend on the variable x_2 and if x_2 is large enough we might have magnitude of a_{11} greater than one. In this case we do not get a minimum-phase prediction-error filter which is sometimes not desirable in practice.

Burg's method estimates the reflection coefficients and then uses the Levinson recursion to estimate the AR parameters. Burg's minimization criteria is to minimize both the forward prediction error $e_p^+(n)$ and the backward prediction error $e_p^-(n)$:

$$\hat{\varphi} = \frac{1}{N-p} \sum_{n=p}^{N-1} \left[e_p^+(n)^2 + e_p^-(n)^2 \right]$$

The computational steps are summarized below [Ref. 7]:

1- Initialize, by setting ;

$$e_0^+(n) = e_0^-(n) = x_n \qquad \text{for} \qquad 0 \le n \le N-1 \tag{3.11}$$

and

$$E_0 = \frac{1}{N} \sum_{n=0}^{N-1} x_n^2$$

At stage p-1, the prediction-error filter will be $A_{p-1}(z)$ which is the Z transform of the sequence $\{1, a_{p-1,1}, a_{p-1,2}, \ldots\ldots, a_{p-1,p-1}\}$. The mean-squared error will be E_{p-1}. $e_{p-1}^-(n)$, and $e_{p-1}^-(n)$ can be calculated for $p-1 \le n \le N-1$.

2- Compute the reflection coefficient:

$$\Upsilon_p = \frac{2 \sum_{n=p}^{N-1} e_{p-1}^+(n) e_{p-1}^-(n-1)}{\sum_{n=p}^{N-1} e_{p-1}^+(n)^2 + e_{p-1}^-(n-1)^2} \tag{3.12}$$

To guarantee the minimum-phase property, Υ_p must have a magnitude less than one.

3- Compute $A_p(z)$ using the Levinson recursion given by:

$$\begin{bmatrix} 1 \\ a_{p,1} \\ a_{p,2} \\ \vdots \\ a_{p,p-1} \\ a_{p,p} \end{bmatrix} = \begin{bmatrix} 1 \\ a_{p-1,1} \\ a_{p-1,2} \\ \vdots \\ a_{p-1,p-1} \\ 0 \end{bmatrix} - \Upsilon_p \begin{bmatrix} 0 \\ a_{p-1,p-1} \\ a_{p-1,p-2} \\ \vdots \\ a_{p-1,1} \\ 1 \end{bmatrix} \tag{3.13}$$

4- Compute $e_p^+(n)$ and $e_p^-(n)$ for $p \le n \le N-1$.

$$e_p^+(n) = e_{p-1}^+(n) - \Upsilon_p e_{p-1}^-(n) \tag{3.14}$$

$$e_p^-(n) = e_{p-1}^-(n-1) - \Upsilon_p e_{p-1}^+(n)$$

5- Update the mean-squared error as follows:

$$E_p = (1 - \Upsilon_p)^2 E_{p-1} \tag{3.15}$$

6- Continue the iteration until p equals the model order.

The Burg's method estimates the poles which are on or inside the unit circle. This is due to the property $|\Upsilon_p| \le 1$. Therefore, care must be taken to deal with the situation when $|\Upsilon_p| = 1$, as this causes the prediction filter to become non-minimum phase.

B. MA PARAMETER ESTIMATION.

The most obvious approach to estimate the MA parameters would be to solve the nonlinear equation (2.19) using the autocorrelation sequence. Solutions of Eq. (2.19) involve difficult spectral factorization techniques [Ref. 8].

There is another approach called *Durbin's method* which is related Kolmogorov's theorem [Ref. 4] and is based on a high order AR approximation of the MA process. The AR process allows results using only linear operations. Let

$$B(z) = 1 + \sum_{k=1}^{q} b_k z^{-1}$$

represent the system transfer function of an MA(q) process, and

$$A_\infty(z) = 1 + \sum_{k=1}^{\infty} a_k z^{-k}$$

represent the system transfer function of an $AR(\infty)$ process that is equivalent to the MA(q) process. Therefore, we have

$$B(z) = \frac{1}{A_\infty(z)} \qquad or$$

$$B(z)A_\infty(z) = 1 \tag{3.16}$$

The inverse Z-transform of the Eq. (3.16) is:

$$a_m + \sum_{n=1}^{q} b_n a_{m-n} = \delta(m) = \begin{cases} 1 & \text{for} \quad m = 0 \\ 0 & \text{for} \quad m = 1,2,\ldots,q \end{cases} \tag{3.17}$$

where $a_0 = 1$ and $a_k = 0$ for $k < 0$. Therefore, the MA parameters can be determined from the infinite-order AR model by solving (3.17).

In practice, one can calculate high-order AR(M) parameters, where $M > q$. Based on these parameter estimates $(1, \hat{a}_M(1), \hat{a}_M(2), \ldots\ldots, \hat{a}_M(M))$, an error in the MA part is computed [Ref. 2].

$$e_{MA}(m) = \hat{a}_M(m) + \sum_{n=1}^{q} b_n \hat{a}_M(m-n) \tag{3.18}$$

According to Eq. (3.17) the error should be zero for all m except for m = 0. But in practice, the error will not be zero when using finite data, so MA parameter estimates are obtained by the minimization of the squared error power, given by

$$\hat{\rho}_q = \sum_{m=0}^{\infty} \frac{|e_{MA}(m)|^2}{M} \tag{3.19}$$

This estimation procedure is an approximate maximum likelihood estimate (MLE). The approximate MLE procedure using Durbin's method for MA parameters results in the following estimation [Ref. 5].

$$\hat{b} = -\hat{R}_{aa}^{-1} \hat{r}_{aa} \tag{3.20}$$

where

$$\left[\hat{R}_{aa}\right]_{ij} = \frac{1}{M+1} \sum_{n=0}^{M-|i-j|} \hat{a}_n \hat{a}_{n+|i-j|} \qquad \text{for} \qquad i,j = 1, 2, \ldots\ldots, q$$

$$\left[\hat{r}_{aa}\right]_i = \frac{1}{M+1} \sum_{n=0}^{M-i} \hat{a}_n \hat{a}_{n+i} \qquad \text{for} \qquad i = 1, 2, \ldots\ldots, q$$

Again the Levinson algorithm may be used to solve Eq. (3.20) for the b parameters. The estimated zeros of B(z) will be inside the unit circle by the minimum-phase property of the autocorrelation method.

In summary, Durbin's method first uses the data $x_0, x_1, ..., x_{N-1}$ to find a large order AR(M) model using the autocorrelation method. Then using these AR parameter estimates $(1, \hat{a}_1, \hat{a}_2, ..., \hat{a}_M)$ as the data, $(\hat{b}_1, \hat{b}_2, ..., \hat{b}_q)$ is found.

Another technique for estimating the MA parameters is to use the MA spectral estimator [Ref. 5] which is given by Eq. (3.21) based on sample correlation values.

$$\hat{P}_{MA}(f) = \sum_{k=-q}^{q} \hat{r}_{xx}(k) e^{-j2\pi fk} \tag{3.21}$$

Since theoretically Eq. (3.21) is equal to $\hat{\sigma}_\varepsilon^2 \left| \hat{B}(f) \right|^2$, the MA parameters can be found by using the Spectral factorization theorem [Ref. 9]. This theorem shows that any rational power spectral density of a stationary signal x_n can be factored into a minimum-phase form

$$P_{xx}(z) = \sigma_\varepsilon^2 B(z) B(z^{-1}) \tag{3.22}$$

C. ITERATIVE ARMA PARAMETER ESTIMATION

Let us consider the modelling of ARMA transfer function as given in Figure 5.

The computational steps are summarized below for the iteration assuming knowledge of the model order:

1- Form the sample correlation of the time series y(n) by using the biased autocorrelation estimator.

2- Get the AR coefficients using the correlation lags > q (to minimize the MA influence).

3- Inverse filter the original time series y(n) via the inverse of the AR filter (use the AR coefficients of step 2) to get $\hat{x}_2(n)$

4- Form the sample correlation of the time series $\hat{x}_2(n)$ by using the biased autocorrelation estimator.

5- Get the MA coefficients using the sample correlation of $\hat{x}_2(n)$.

6- Inverse filter the original time series y(n) via the inverse of the MA filter (use MA coefficients which are estimated in step 5) to get $\hat{x}_1(n)$. We assume that the MA coefficients are of minimum phase (all roots are inside the unit circle).

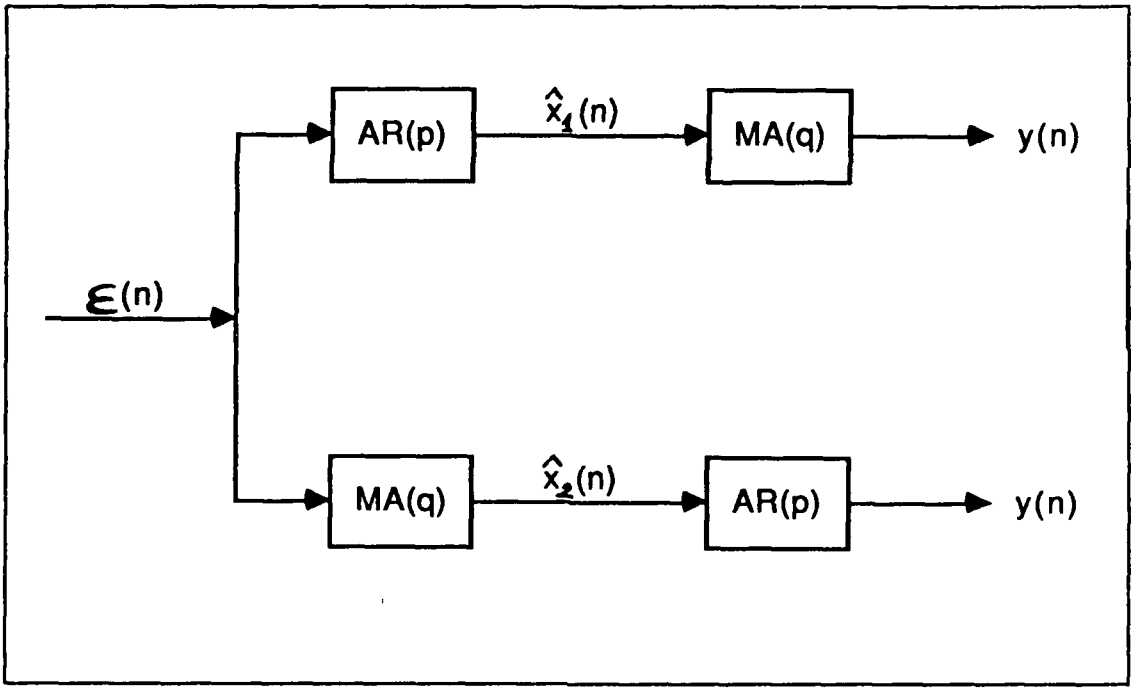

Figure 5. Modelling of ARMA transfer function.

7- Form the sample correlation of time series $\hat{x}_1(n)$ by using the biased autocorrelation estimator.

8- Get the AR coefficients using all lags or $lags > q$ depending on the method used.

9- Inverse filter the original y(n) series using the AR coefficient estimates which are obtained in step 8, to get $\hat{x}_2(n)$.

10- Get the sample correlation of $\hat{x}_2(n)$ by using the biased autocorrelation estimator.

11- Compute the MA coefficients using the sample correlation of $\hat{x}_2(n)$.

12- Compute the error in the AR and the MA parts as given below

$$error(j-1) = \sum_{i=1}^{p} \left(a_i^j - a_i^{j-1}\right)^2 + \sum_{i=1}^{q} \left(b_i^j - b_i^{j-1}\right)^2 \quad \text{for } j = 2,3,\ldots,10 \quad (3.23)$$

where superscript j is used to denote the number of iteration. The upper count of j is experimentally chosen. Iteration continue up to 10 if the coefficients do not converge.

13- If $error > \lambda$, then go to step 6, else exit the program using the last updates of the filter coefficients. If j > 10 terminate with an error message. Note λ is a small experimentally chosen number.

The AR coefficients can be obtained via a pseudoinverse from the modified Yule-Walker equations [Ref. 2]. Equation (2.17) may be rewritten for the p lag indices $q+1 \le m \le q+p$, and put into matrix form

$$\begin{bmatrix} r_{xx}(q) & r_{xx}(q-1) & r_{xx}(q-p+1) \\ r_{xx}(q+1) & r_{xx}(q) & r_{xx}(q-p+2) \\ \vdots & \vdots & \vdots \\ r_{xx}(q+p+1) & r_{xx}(q+p+2) & r_{xx}(q) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = - \begin{bmatrix} r_{xx}(q+1) \\ r_{xx}(q+2) \\ \vdots \\ r_{xx}(q+p) \end{bmatrix} \quad (3.24)$$

We can compute the autocorrelation sequence for lags q-p+1 to q+p, therefore the AR parameters are found as the solution of Eq. (3.24) where the MA parts influence is minimal. The autocorrelation matrix is of Toeplitz form.

The Moore-Penrose pseudoinverse $A^{\#}$ of an $m \times n$ matrix A provides the *minimum-norm least squares solution* to $Ax = b$. The solution is given by

$$x = A^{\#} b$$

where x is a $n \times 1$ vector that simultaneously minimizes the squared equation error, for a given $m \times 1$ vector b [Ref. 2]. If m = n and rank A is n (i.e., A is nonsingular), then the pseudoinverse becomes the square matrix inverse $A^{\#} = A^{-1}$. If $m > n$ (i.e., more equations than unknowns) and rank A is n, then

$$A^{\#} = (A^H A)^{-1} A^H \qquad \textit{and}$$

$$x = (A^H A)^{-1} A^H b$$

The superscript H is used to denote the Hermitian transpose operation. This is the least squares solution for a set of overdetermined equations.

In step 2 of the iteration, we will use correlation lags greater than q while in step 8 either all lags or lags greater than q are used depending on the method.

For MA coefficients estimation the Cholesky decomposition is used. If the matrix A is square and Hermitian, then the usual triangular factorization takes on the special form

$$A = RR^H \quad (3.25)$$

where R is a lower triangular matrix with nonzero real principal diagonal elements. This decomposition is called the *Cholesky decomposition* [Ref. 2].

For the MA part, the statistical autocorrelation is given by

$$R_{xx}(k) = \sigma_\varepsilon^2 \sum_{n=0}^{q-|k|} b_{n+k} b_n \tag{3.26}$$

Let B be the lower triangular Toeplitz matrix where the matrix elements b_{nl} are given in terms of the impulse response of the filter B(z) [Ref. 7]:

$$b_{nl} = b_{n-l}$$

and let the autocorrelation matrix of x_n be

$$R_{xx}(i,j) = R_{xx}(i-j)$$

Then, the transpose matrix B^H will have matrix elements

$$(B^H)_{ni} = b_{i-n}$$

and Eq. (3.26) can be written as

$$R_{xx}(i,j) = R_{xx}(i-j) = \sigma_\varepsilon^2 \sum_{n=0}^{q-|i-j|} b_{n+i-j} b_n$$

$$R_{xx}(i,j) = \sigma_\varepsilon^2 (BB^H)_{ij}$$

Thus, in matrix notation

$$R_{xx} = \sigma_\varepsilon^2 BB^H \tag{3.27}$$

which is related to Eq. (3.25) with the assumption that

$$\sigma_\varepsilon^2 = 1$$

Therefore, an approximation of the MA parameters can be found from the Cholesky decomposition of the correlation matrix R_{xx}.

IV. SIMULATION RESULTS

In this chapter, computer simulation results are presented to show how three different estimation methods work on various ARMA models. Two ARMA(2,2) models which differ in pole-zero locations, an ARMA(2,3) model and an ARMA(3,4) model are used as test models. In addition two realizations for each ARMA process are utilized. Two hundred data points are used in computing the sample autocorrelation values. The three different AR estimation methods are explained below.

Method 1

The AR coefficients are obtained via a square matrix inverse using the modified Yule-Walker equations. Correlation lags greater than q are used to minimize the MA part influence at the first calculation. For the remainder of the iteration the same correlation lags are used to minimize potential influence from the MA part. A Cholesky factorization is used to find the MA part coefficients.

Method 2

The AR coefficients are obtained via a square matrix inverse using the modified Yule-Walker equations. Correlation lags greater than q are used to minimize the MA part influence at first calculation. For the remainder of the iteration the correlation lags starting from zero are used assuming the MA part contribution has effectively been removed. The Cholesky factorization is used to find the MA part coefficients.

Method 3

The AR coefficients are obtained via a pseudoinverse instead of a square matrix inverse using the correlation function starting at the zero lag after first calculation. This allows the use of all important correlation lags. The Cholesky factorization is used to find the MA part coefficients.

In addition, observation noise is added to one of the ARMA models and the resulting noisy sequence is processed via the three different methods. The observation noise is independent of the driving noise. It is white Gaussian noise with zero mean and unit variance. The signal to noise ratio (SNR) is about 15 dB.

The computer program computes the differential errors in the AR and MA parts as given in Eq.(3.23), and then compares it with a small experimentally established value λ (i.e., $\lambda = 0.0001$). If the error is less than λ the program is terminated. If the error is

larger than λ the program is reentered at step 6. Also, if the coefficients do not converge the program will stop after nine iterations.

Comparisons of the true and estimated coefficient differences, of pole-zero locations, of distances between the true and estimated pole-zero locations and of radial differences between the true and estimated pole-zero locations are presented to show how well the three methods work. Also the spectra of the ARMA models are plotted by using the true and estimated coefficients.

A. THE ARMA(2,2) MODEL-A

The pole-zero locations for this model are illustrated in Figure 6.

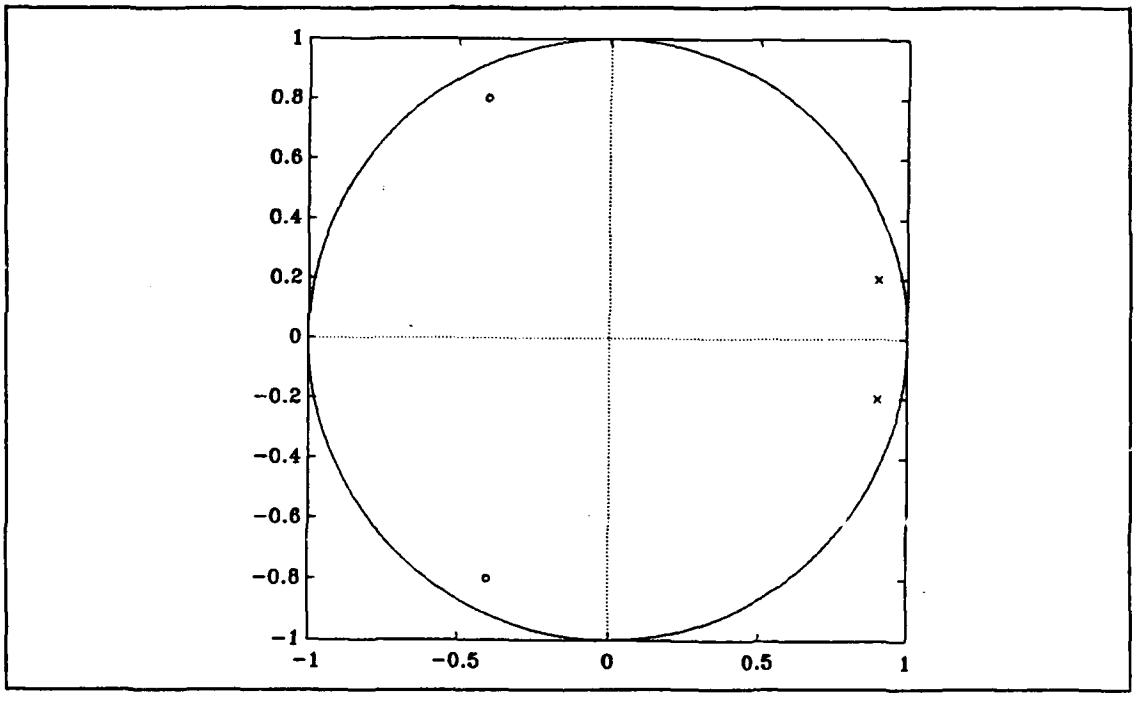

Figure 6. The ARMA(2,2) Model-A, Pole-zero Locations

1. METHOD 1

a. Noise Realization 1

The coefficients converge after two iterations. Tables 1 and 2 present the results.

Table 1. ARMA(2,2) MODEL-A, METHOD 1, COEFFICIENTS COMPARISON, REALIZATION 1.

Coeff.	True	Estimated	Difference
a_1	-1.80	-1.8412	-0.0412
a_2	0.85	0.8972	+0.0472
b_0	1.00	1.3053	+0.3053
b_1	0.80	0.6439	-0.1561
b_2	0.80	0.0572	-0.7428

Table 2. ARMA(2,2) MODEL-A, METHOD 1, POLE-ZERO COMPARISON, REALIZATION 1.

Poles-Zeros	True	Estimated	Distance	Radial Diff.
p_1	0.9 + 0.2j	0.9206 + 0.2230j	0.0308	0.0189
p_2	0.9 - 0.2j	0.9206 - 0.2230j	0.0308	0.0189
z_1	-0.4 + 0.8j	-0.3771	0.8003	1.1071
z_2	-0.4 - 0.8j	-0.1162	0.8488	1.1071

b. Noise Realization 2

Using a different noise realization, the coefficients still converge after two iterations. Tables 3 and 4 present the results.

Table 3. ARMA(2,2) MODEL-A, METHOD 1, COEFFICIENTS COMPARISON, REALIZATION 2

Coeff.	True	Estimated	Difference
a_1	-1.80	-1.8712	-0.0712
a_2	0.85	0.9167	+0.0667
b_0	1.00	1.1083	+0.1083
b_1	0.80	0.5071	-0.2929
b_2	0.80	0.1502	-0.6498

Table 4. ARMA(2,2) MODEL-A, METHOD 1, POLE-ZERO COMPARISON, REALIZATION 2

Poles-Zeros	True	Estimated	Distance	Radial Diff.
p_1	0.9 + 0.2j	0.9356 + 0.2033j	0.0358	0.0047
p_2	0.9 - 0.2j	0.9356 - 0.2033j	0.0358	0.0047
z_1	-0.4 + 0.8j	-0.2288 + 0.2884j	0.5394	0.2070
z_2	-0.4 - 0.8j	-0.2288 - 0.2884j	0.5394	0.2070

The Spectra using the true and the estimated network coefficients are plotted in Figure 7.

2. METHOD 2

a. Noise Realization 1

The coefficients converge after two iterations. Tables 5 and 6 present the results.

Table 5. ARMA(2,2) MODEL-A, METHOD 2, COEFFICIENTS COMPARISON, REALIZATION 1.

Coeff.	True	Estimated	Difference
a_1	-1.80	-2.0065	-0.2065
a_2	0.85	1.0544	+0.2044
b_0	1.00	1.3866	+0.3866
b_1	0.80	0.6467	-0.1533
b_2	0.80	-0.0178	-0.8178

Table 6. ARMA(2,2) MODEL-A, METHOD 2, POLE-ZERO COMPARISON, REALIZATION 1.

Poles-Zeros	True	Estimated	Distance	Radial Diff.
p_1	0.9 + 0.2j	1.0032 + 0.2189j	0.1049	0.0038
p_2	0.9 - 0.2j	1.0032 - 0.2189j	0.1049	0.0038
z_1	-0.4 + 0.8j	-0.4924	0.8053	1.1071
z_2	-0.4 - 0.8j	0.0261	0.9064	2.0344

b. Noise Realization 2

Using a different noise realization, the coefficients still converge after two iterations. Tables 7 and 8 present the results.

Table 7. ARMA(2,2) MODEL-A, METHOD 2, COEFFICIENTS COMPARISON, REALIZATION 2.

Coeff.	True	Estimated	Difference
a_1	-1.80	-1.9267	-0.1267
a_2	0.85	0.9729	+0.1229
b_0	1.00	1.0982	+0.0982
b_1	0.80	0.4598	-0.3402
b_2	0.80	0.0902	-0.7098

Table 8. ARMA(2,2) MODEL-A, METHOD 2, POLE-ZERO COMPARISON, REALIZATION 2.

Poles-Zeros	True	Estimated	Distance	Radial Diff.
p_1	0.9 + 0.2j	0.9634 + 0.2118j	0.0644	0.0022
p_2	0.9 - 0.2j	0.9634 - 0.2118j	0.0644	0.0022
z_1	-0.4 + 0.8j	-0.2094 + 0.1958j	0.6335	0.3553
z_2	-0.4 - 0.8j	-0.2094 - 0.1958j	0.6335	0.3553

The spectra using the true and the estimated network coefficients are plotted in Figure 8.

3. METHOD 3

a. Noise Realization 1

The coefficients converge after two iterations. Tables 9 and 10 present the results.

Table 9. ARMA(2,2) MODEL-A, METHOD 3, COEFFICIENTS COMPARISON, REALIZATION 1.

Coeff.	True	Estimated	Difference
a_1	-1.80	-1.8375	-0.0375
a_2	0.85	0.8948	+0.0448
b_0	1.00	1.3062	+0.3062
b_1	0.80	0.6482	-0.1518
b_2	0.80	0.0657	-0.7343

Table 10. ARMA(2,2) MODEL-A, METHOD 3, POLE-ZERO COMPARISON, REALIZATION 1.

Poles-Zeros	True	Estimated	Distance	Radial Diff.
p_1	0.9 + 0.2j	0.9187 + 0.2253j	0.0314	0.0218
p_2	0.9 - 0.2j	0.9187 - 0.2253j	0.0314	0.0218
z_1	-0.4 + 0.8j	-0.3542	0.8013	1.1071
z_2	-0.4 - 0.8j	-0.1421	0.8405	1.1071

b. Noise Realization 2

For a different noise realization, the coefficients still converge after one iteration. Tables 11 and 12 present the results.

Table 11. ARMA(2,2) MODEL-A, METHOD 3, COEFFICIENTS COMPARISON, REALIZATION 2

Coeff.	True	Estimated	Difference
a_1	-1.80	-1.8545	-0.0545
a_2	0.85	0.9043	+0.0543
b_0	1.00	1.1144	+0.1144
b_1	0.80	0.5272	-0.2728
b_2	0.80	0.1808	-0.6192

Table 12. ARMA(2,2) MODEL-A, METHOD 3, POLE-ZERO COMPARISON, REALIZATION 2

Poles-Zeros	True	Estimated	Distance	Radial Diff.
p_1	0.9 + 0.2j	0.9272 + 0.2111j	0.0293	0.0051
p_2	0.9 - 0.2j	0.9272 - 0.2111j	0.0293	0.0051
z_1	-0.4 + 0.8j	-0.2365 + 0.3260j	0.5014	0.1639
z_2	-0.4 - 0.8j	-0.2365 - 0.3260j	0.5014	0.1639

The spectra using the true and the estimated network coefficients are plotted in Figure 9.

In summary, the estimated coefficients converge after two iterations in all three methods. For both noise realizations, the third method gives the best result in terms of the coefficients and pole-zero locations. For the noise realization 2, the coefficients converge after one iteration using method 3.

In each method the AR part coefficients of the ARMA(2,2) model tend to be more accurate than the MA part coefficients.

B. THE ARMA(2,2) MODEL-B

The pole-zero locations for this model are illustrated in Figure 10.

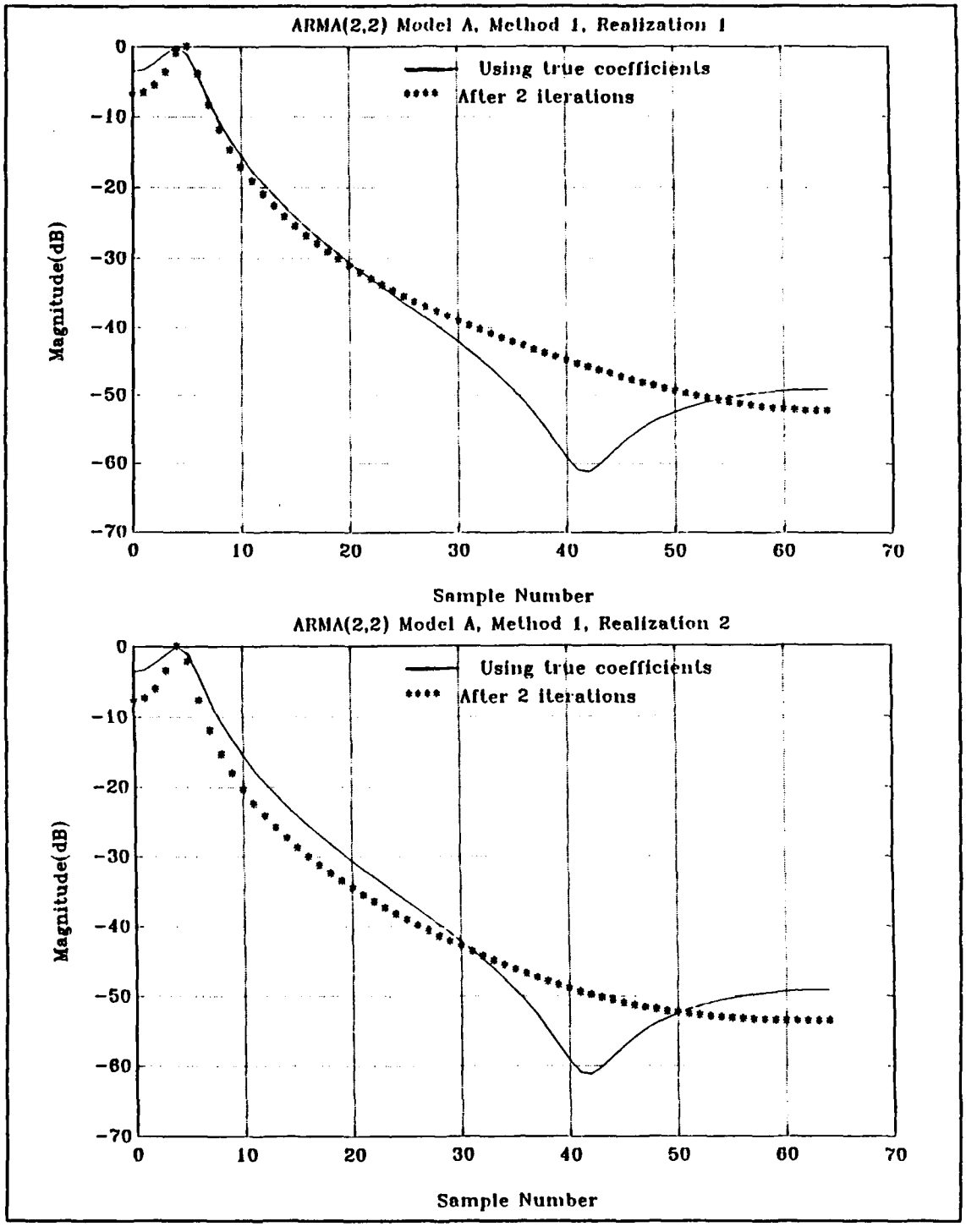

Figure 7. The Spectra of ARMA(2,2) Model-A, Method 1

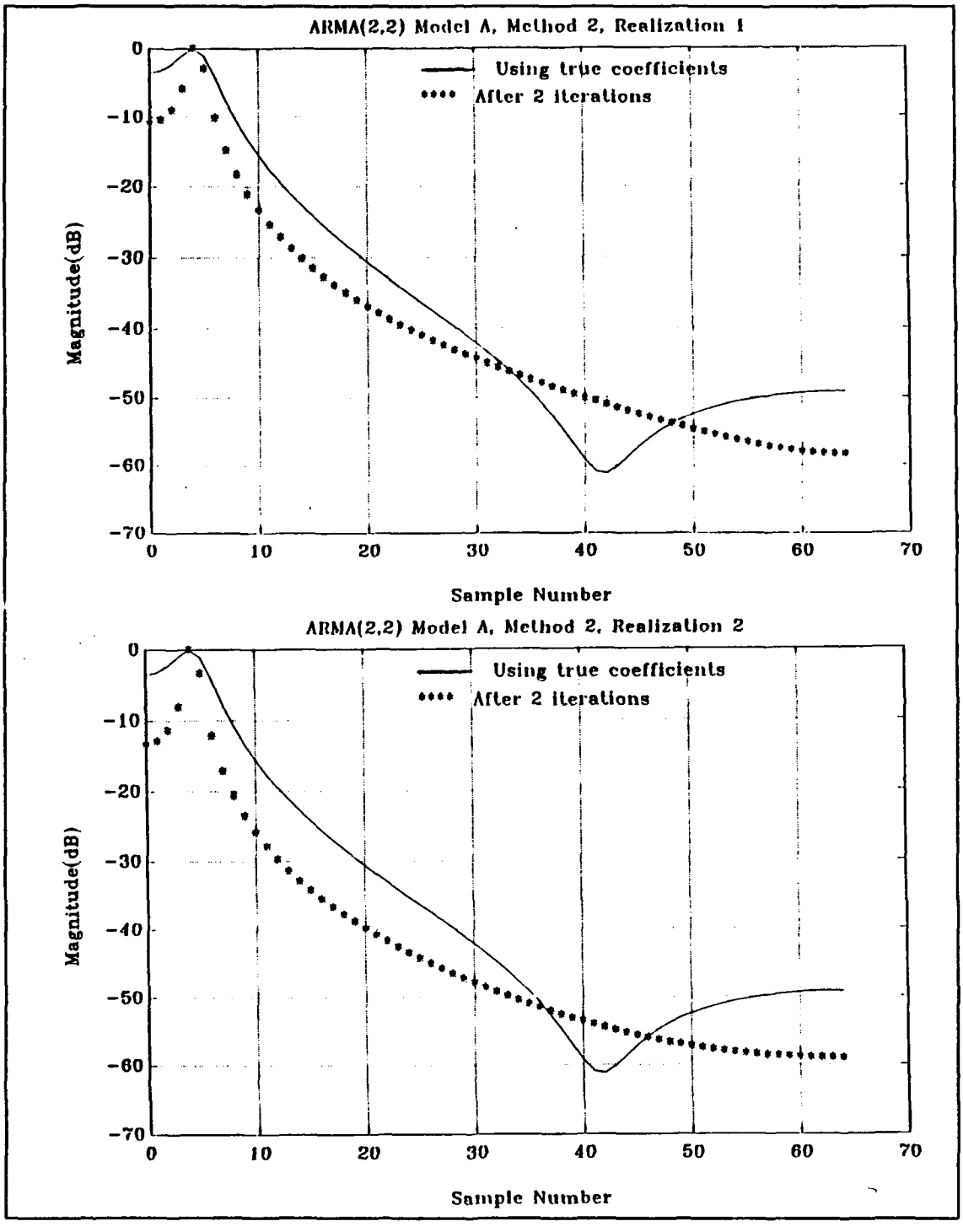

Figure 8. The Spectra of ARMA(2,2) Model-A, Method 2

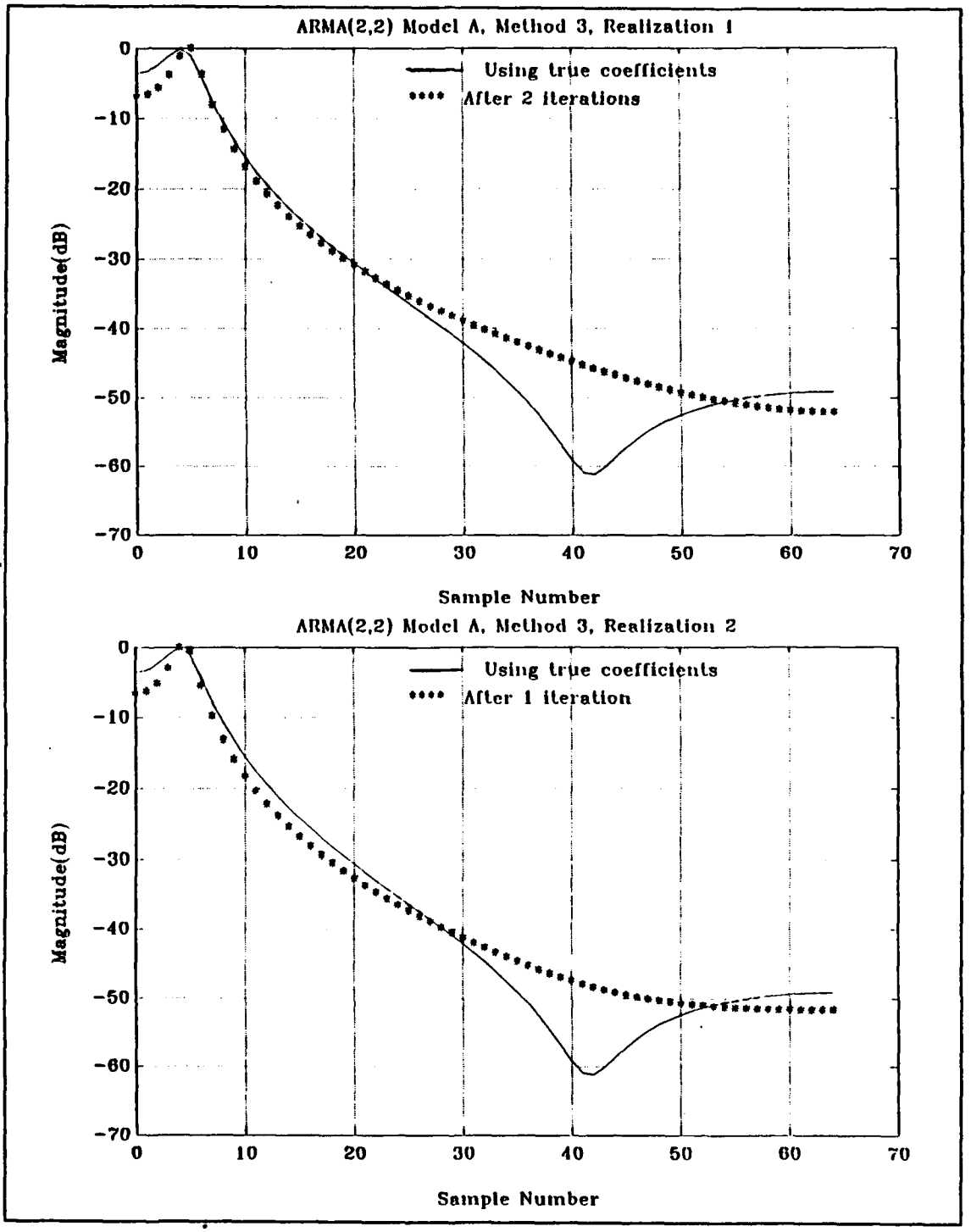

Figure 9. The Spectra of ARMA(2,2) Model-A, Method 3

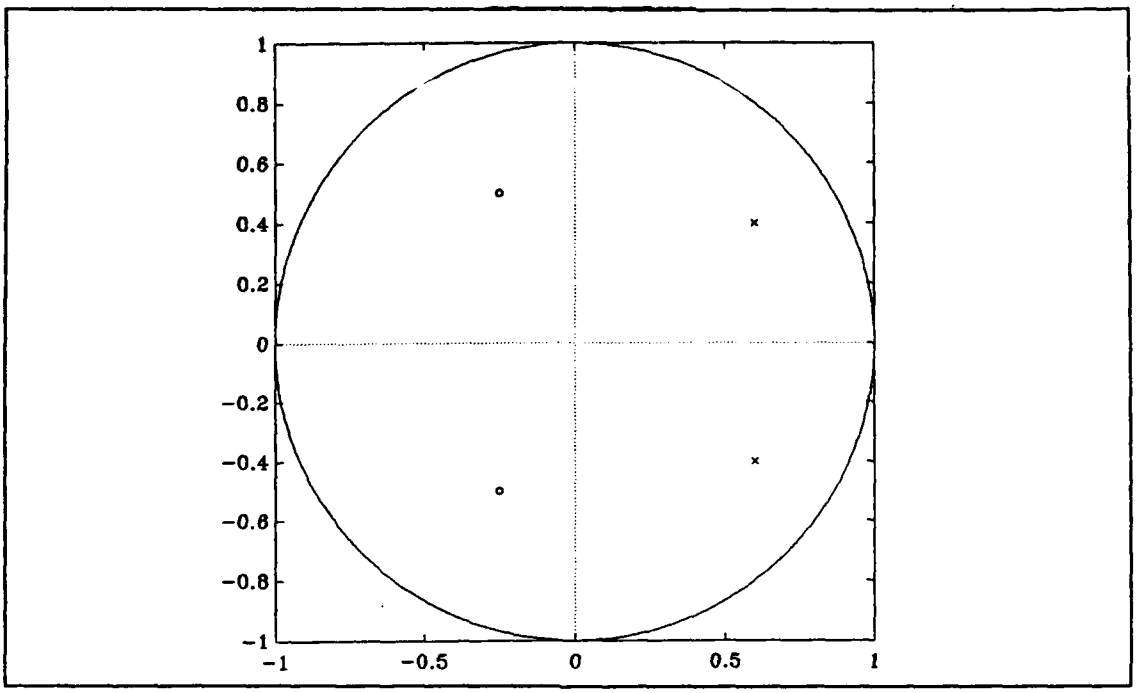

Figure 10. The ARMA(2,2) Model-B, Pole-zero Locations

1. METHOD 1

a. Noise Realization 1

The coefficients converge after three iterations. Tables 13 and 14 present the results.

Table 13. ARMA(2,2) MODEL-B, METHOD 1, COEFFICIENTS COMPARISON, REALIZATION 1.

Coeff.	True	Estimated	Difference
a_1	-1.20	-1.2614	-0.0614
a_2	0.52	0.4143	-0.1057
b_0	1.00	1.0518	+0.0518
b_1	0.50	0.4045	-0.0955
b_2	0.3125	-0.1317	-0.4442

Table 14. ARMA(2,2) MODEL-B, METHOD 1, POLE-ZERO COMPARISON, REALIZATION 1.

Poles-Zeros	True	Estimated	Distance	Radial Diff.
p_1	0.6 + 0.4j	0.6307 + 0.1285j	0.2732	0.3870
p_2	0.6 - 0.4j	0.6307 - 0.1285j	0.2732	0.3870
z_1	-0.25 + 0.5j	-0.5950	0.6074	1.1071
z_2	-0.25 - 0.5j	0.2104	0.6796	2.0344

b. Noise Realization 2

For a different noise realization, the coefficients converge after two iterations. Tables 15 and 16 present the results.

Table 15. ARMA(2,2) MODEL-B, METHOD 1, COEFFICIENTS COMPARISON, REALIZATION 2

Coeff.	True	Estimated	Difference
a_1	-1.20	-1.6704	-0.4704
a_2	0.52	0.7879	+0.2679
b_0	1.00	1.1018	+0.1018
b_1	0.50	0.1188	-0.3812
b_2	0.3125	-0.3508	-0.6633

Table 16. ARMA(2,2) MODEL-B, METHOD 1, POLE-ZERO COMPARISON, REALIZATION 2

Poles-Zeros	True	Estimated	Distance	Radial Diff.
p_1	0.6 + 0.4j	0.8352 + 0.3006j	0.2553	0.2425
p_2	0.6 - 0.4j	0.8352 - 0.3006j	0.2553	0.2425
z_1	-0.25 + 0.5j	-0.6207	0.6224	1.1071
z_2	-0.25 - 0.5j	0.5129	0.9121	2.0344

The spectra using the true and the estimated network coefficients are plotted in Figure 11.

2. METHOD 2

a. Noise Realization 1

The coefficients converge after six iterations. Tables 17 and 18 present the results.

Table 17. ARMA(2,2) MODEL-B, METHOD 2, COEFFICIENTS COMPARISON, REALIZATION 1.

Coeff.	True	Estimated	Difference
a_1	-1.20	-1.1531	+0.0469
a_2	0.52	0.5013	-0.0187
b_0	1.00	1.0174	+0.0174
b_1	0.50	0.4732	-0.0268
b_2	0.3125	0.1202	-0.1923

Table 18. ARMA(2,2) MODEL-B, METHOD 2, POLE-ZERO COMPARISON, REALIZATION 1.

Poles-Zeros	True	Estimated	Distance	Radial Diff.
p_1	0.6 + 0.4j	0.5766 + 0.4110j	0.0258	0.0312
p_2	0.6 - 0.4j	0.5766 - 0.4110j	0.0258	0.0312
z_1	-0.25 + 0.5j	-0.2326 + 0.2532j	0.2474	0.2793
z_2	-0.25 - 0.5j	-0.2326 - o.2532j	0.2474	0.2793

b. Noise Realization 2

For a different noise realization, the coefficients converge after five iterations. Tables 19 and 20 present the results.

Table 19. ARMA(2.2) MODEL-B, METHOD 2, COEFFICIENTS COMPARISON, REALIZATION 2

Coeff.	True	Estimated	Difference
a_1	-1.20	-1.5914	-0.3914
a_2	0.52	0.7540	+0.2340
b_0	1.00	1.0610	+0.0610
b_1	0.50	0.1301	-0.3699
b_2	0.3125	-0.2894	-0.6019

Table 20. ARMA(2,2) MODEL-B, METHOD 2, POLE-ZERO COMPARISON, REALIZATION 2

Poles-Zeros	True	Estimated	Distance	Radial Diff.
p_1	0.6 + 0.4j	0.7957 + 0.3477j	0.2025	0.1760
p_2	0.6 - 0.4j	0.7957 - 0.3477j	0.2025	0.1760
z_1	-0.25 + 0.5j	-0.5872	0.6030	1.1071
z_2	-0.25 - 0.5j	0.4645	0.8720	2.0344

The spectra using the true and the estimated network coefficients are plotted in Figure 12.

3. METHOD 3

a. Noise Realization 1

The coefficients converge after four iterations. Tables 21 and 22 present the results.

Table 21. ARMA(2,2) MODEL-B, METHOD 3, COEFFICIENTS COMPARISON, REALIZATION 1.

Coeff.	True	Estimated	Difference
a_1	-1.20	-1.1818	+0.0182
a_2	0.52	0.5245	+0.0045
b_0	1.00	1.0030	+0.0030
b_1	0.50	0.4436	-0.0564
b_2	0.3125	0.1015	-0.2110

Table 22. ARMA(2,2) MODEL-B, METHOD 3, POLE-ZERO COMPARISON, REALIZATION 1.

Poles-Zeros	True	Estimated	Distance	Radial Diff.
p_1	0.6 + 0.4j	0.5909 + 0.4187j	0.0207	0.0284
p_2	0.6 - 0.4j	0.5909 - 0.4187j	0.0207	0.0284
z_1	-0.25 + 0.5j	-0.2211 + 0.2286j	0.2729	0.3050
z_2	-0.25 - 0.5j	-0.2211 - 0.2286j	0.2729	0.3050

b. Noise Realization 2

For a different noise realization, the coefficients converge after two iterations. Tables 23 and 24 present the results.

Table 23. ARMA(2,2) MODEL-B, METHOD 3, COEFFICIENTS COMPARISON, REALIZATION 2.

Coeff.	True	Estimated	Difference
a_1	-1.20	-1.5105	-0.3105
a_2	0.52	0.6962	+0.1762
b_0	1.00	1.0488	+0.0488
b_1	0.50	0.1827	-0.3173
b_2	0.3125	-0.2154	-0.5279

Table 24. ARMA(2,2) MODEL-B, METHOD 3, POLE-ZERO COMPARISON, REALIZATION 2.

Poles-Zeros	True	Estimated	Distance	Radial Diff.
p_1	0.6 + 0.4j	0.7553 + 0.3547j	0.1617	0.1489
p_2	0.6 - 0.4j	0.7553 - 0.3547j	0.1617	0.1489
z_1	-0.25 + 0.5j	-0.5486	0.5823	1.1071
z_2	-0.25 - 0.5j	0.3744	0.7999	2.0344

The spectra using the true and the estimated network coefficients are plotted in Figure 13.

In summary, under each noise realization the method 3 gives the best result in terms of the coefficients and pole-zero locations.

For each method, the spectrum of the ARMA(2,2) model using estimated coefficients indicates reasonably accurate pole locations but the zero locations of the spectrum do not follow the original ones. This means that the AR part coefficients of the ARMA model tend to be more accurate than the MA part coefficients.

C. THE ARMA(3,4) MODEL

The pole-zero locations for this model are illustrated in Figure 14.

1. METHOD 1

a. Noise Realization 1

The coefficients do not converge in nine iterations. Because of oscillations about two values, the average of the two values is used as an estimate of the coefficients in the comparison. Tables 25 and 26 present the results.

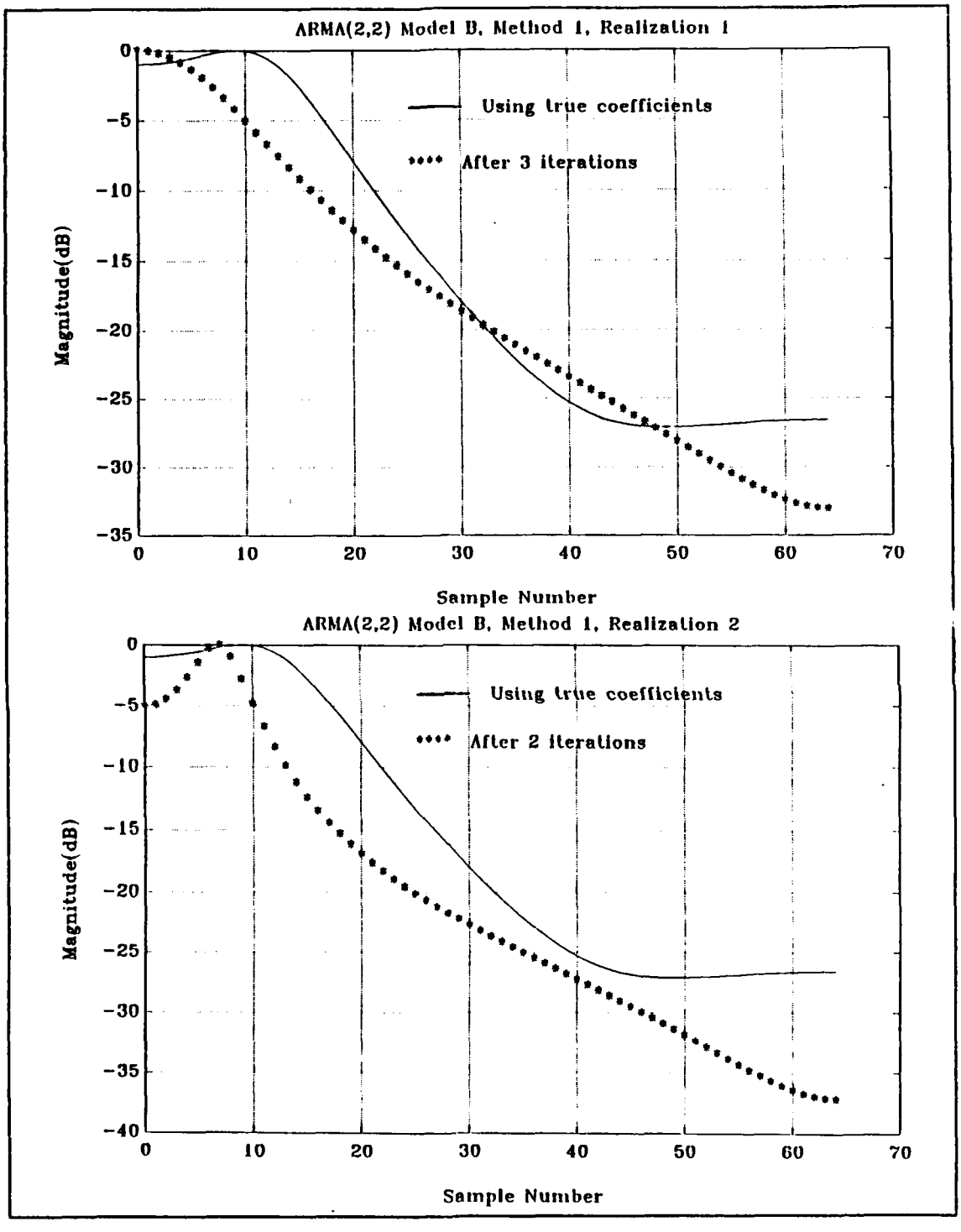

Figure 11. The Spectra of ARMA(2,2) Model-B, Method 1

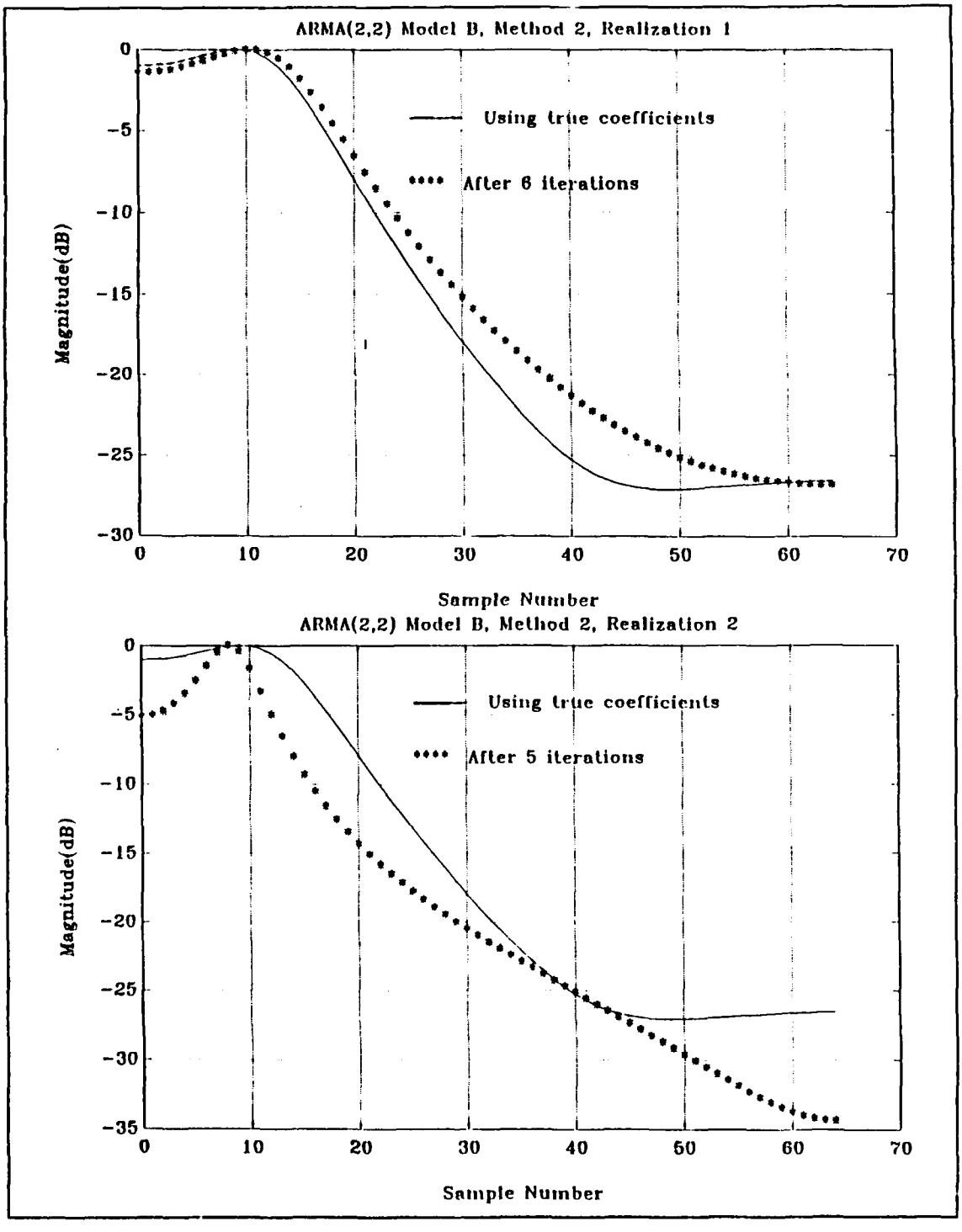

Figure 12. The Spectra of ARMA(2,2) Model-B, Method 2

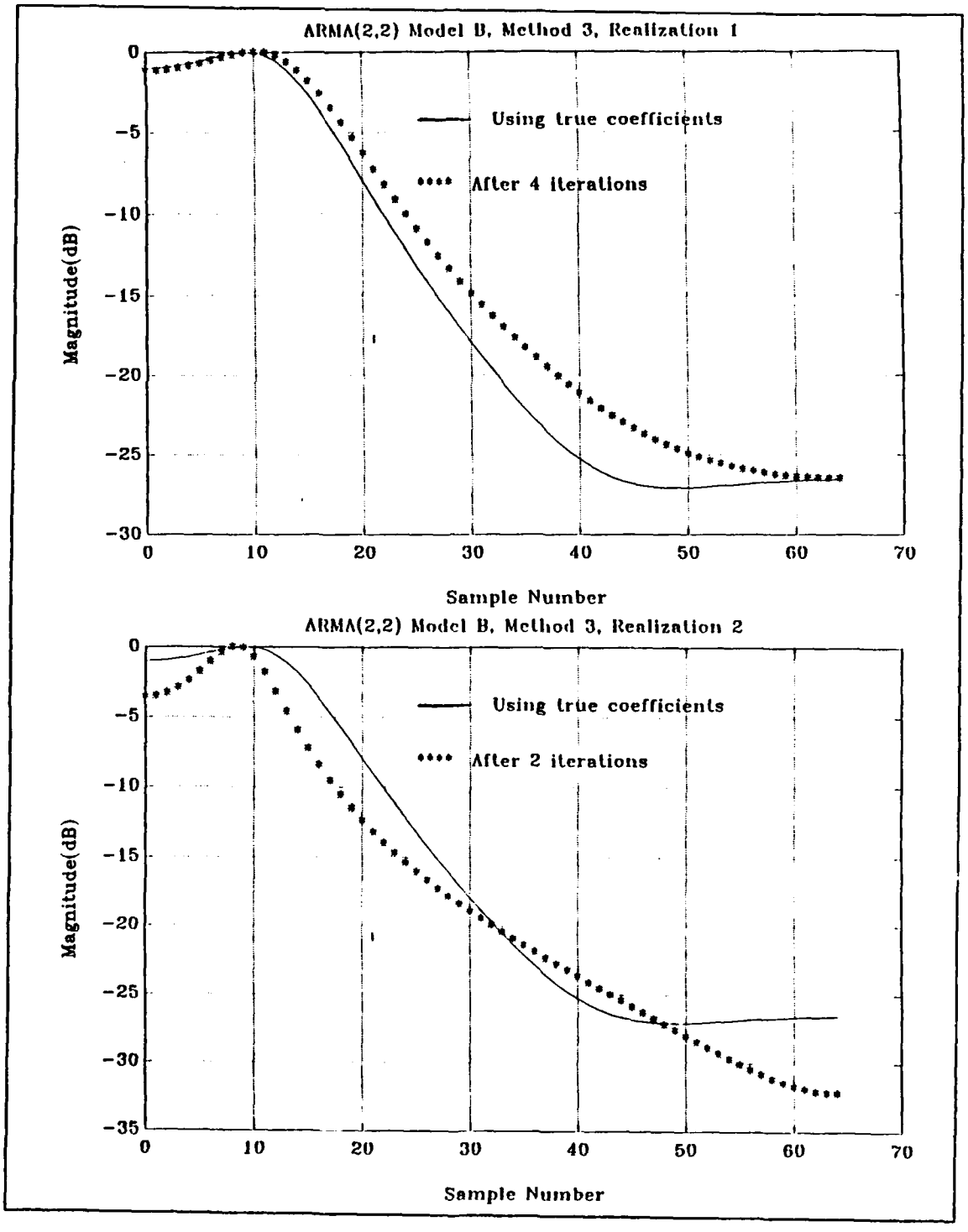

Figure 13. The Spectra of ARMA(2,2) Model-B, Method 3

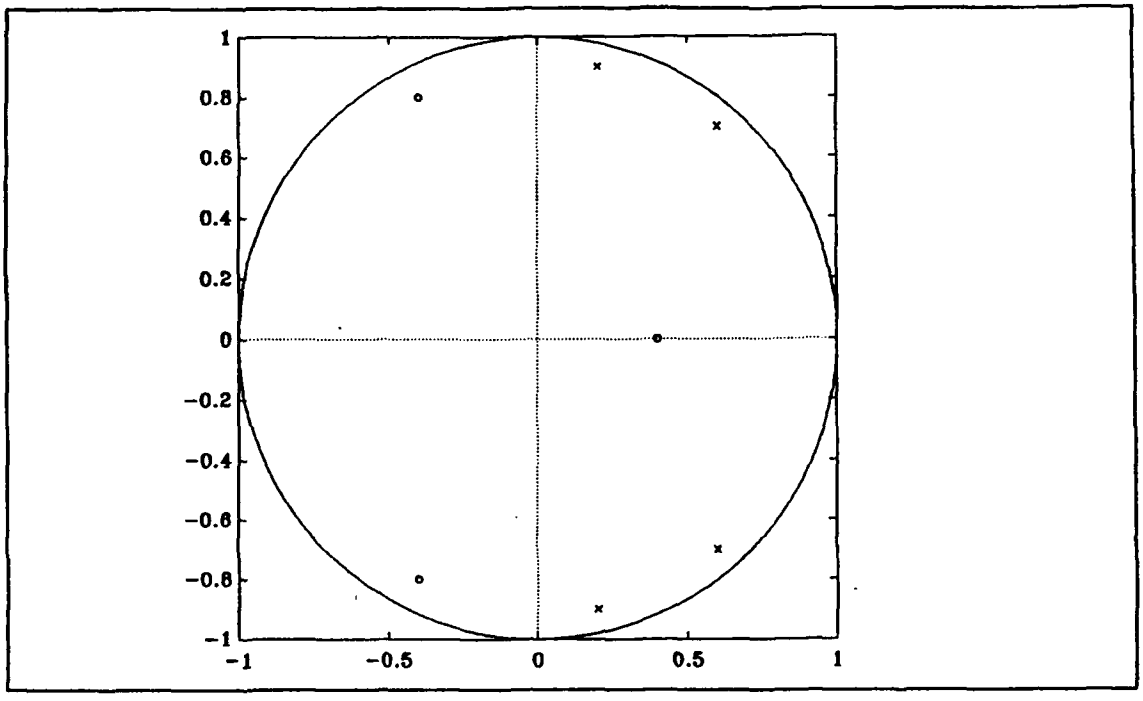

Figure 14. The ARMA(3,4) Model Pole-zero Locations

Table 25. ARMA(3,4) MODEL, METHOD 1, COEFFICIENTS COMPARISON, REALIZATION 1.

Coeff.	True	Estimated	Difference
a_1	-1.60	-1.5275	+0.0725
a_2	2.18	1.5642	-0.6158
a_3	-1.36	-1.0165	+0.3435
a_4	0.7225	0.4656	-0.2569
b_0	1.00	3.3059	+2.3059
b_1	0.40	1.3473	0.9473
b_2	0.48	-1.6482	-2.1282
b_3	-0.32	-2.4187	-2.0987

Table 26. ARMA(3,4) MODEL, METHOD 1, POLE-ZERO COMPARISON, REALIZATION 1.

Poles-Zeros	True	Estimated	Distance	Radial Diff.
p_1	0.2 + 0.9j	0.0718 + 0.8101j	0.1565	0.1302
p_2	0.2 - 0.9j	0.0718 - 0.8101j	0.1565	0.1302
p_3	0.6 + 0.7j	0.6919 + 0.4745j	0.2435	0.2610
p_4	0.6 - 0.7j	0.6919 - 0.4745j	0.2435	0.2610
z_1	-0.4 + 0.8j	-0.6754 + 0.5652j	0.3619	0.4103
z_2	-0.4 - 0.8j	-0.6754 - 0.5652j	0.3619	0.4103
z_3	0.4	0.9433	0.5433	0.0000

b. Noise Realization 2

Using a different noise realization, the coefficients still do not converge in nine iterations. Because of oscillations about two values, the average of the two values is used as an estimate of the coefficients in the comparison. Tables 27 and 28 present the results.

Table 27. ARMA(3,4) MODEL, METHOD 1, COEFFICIENTS COMPARISON, REALIZATION 2.

Coeff.	True	Estimated	Difference
a_1	-1.60	-2.0479	-0.4479
a_2	2.18	2.2690	+0.0890
a_3	-1.36	-1.4743	-0.1143
a_4	0.7225	0.9468	+0.2243
b_0	1.00	2.6176	+1.6176
b_1	0.40	0.8737	+0.4737
b_2	0.48	-1.2400	-1.7200
b_3	-0.32	-1.7033	-1.3833

Table 28. ARMA(3,4) MODEL, METHOD 1, POLE-ZERO COMPARISON, REALIZATION 2.

Poles-Zeros	True	Estimated	Distance	Radial Diff.
p_1	0.2 + 0.9j	0.0518 + 0.8258j	0.1657	0.1560
p_2	0.2 - 0.9j	0.0518 - 0.8258j	0.1657	0.1560
p_3	0.6 + 0.7j	0.9722 + 0.6618j	0.3741	0.2644
p_4	0.6 - 0.7j	0.9722 - 0.6618j	0.3741	0.2644
z_1	-0.4 + 0.8j	-0.6316 + 0.5489j	0.3415	0.3916
z_2	-0.4 - 0.8j	-0.6316 - 0.5489j	0.3415	0.3916
z_3	0.4	0.9294	0.5294	0.0000

The spectra using the true and the estimated network coefficients are plotted in Figure 15.

2. METHOD 2

a. Noise Realization 1

The coefficients converge after nine iterations. Tables 29 and 30 present the results.

Table 29. ARMA(3,4) MODEL, METHOD 2, COEFFICIENTS COMPARISON.REALIZATION 1.

Coeff.	True	Estimated	Difference
a_1	-1.60	-1.6186	-0.0186
a_2	2.18	2.2093	+0.0293
a_3	-1.36	-1.3948	-0.0348
a_4	0.7225	0.7366	+0.0141
b_0	1.00	1.1463	+0.1463
b_1	0.40	0.2664	-0.1336
b_2	0.48	-0.0262	-0.5062
b_3	-0.32	-0.3386	-0.0186

Table 30. ARMA(3,4) MODEL, METHOD 2, POLE-ZERO COMPARISON, REALIZATION 1.

Poles-Zeros	True	Estimated	Distance	Radial Diff.
p_1	0.2 + 0.9j	0.2033 + 0.9077j	0.0083	0.0016
p_2	0.2 - 0.9j	0.2033 - 0.9077j	0.0083	0.0016
p_3	0.6 + 0.7j	0.6060 + 0.6957j	0.0073	0.0079
p_4	0.6 - 0.7j	0.6060 - 0.6957j	0.0073	0.0079
z_1	-0.4 + 0.8j	-0.4197 + 0.5572j	0.2435	0.1819
z_2	-0.4 - 0.8j	-0.4197 - 0.5572j	0.2435	0.1819
z_3	0.4	0.6070	0.2070	0.0000

b. Noise Realization 2

Using the second noise realization, the coefficients converge after nine iterations. Tables 31 and 32 present the results.

Table 31. ARMA(3,4) MODEL, METHOD 2, COEFFICIENTS COMPARISON, REALIZATION 2.

Coeff.	True	Estimated	Difference
a_1	-1.60	-1.6668	-0.0668
a_2	2.18	2.3645	+0.1845
a_3	-1.36	-1.5282	-0.1682
a_4	0.7225	0.8527	+0.1302
b_0	1.00	0.9923	-0.0077
b_1	0.40	0.1274	-0.2726
b_2	0.48	0.2222	-0.2578
b_3	-0.32	-0.1221	+0.1979

Table 32. ARMA(3,4) MODEL, METHOD 2, POLE-ZERO COMPARISON, REALIZATION 2.

Poles-Zeros	True	Estimated	Distance	Radial Diff.
p_1	0.2 + 0.9j	0.2063 + 0.9315j	0.0321	0.0007
p_2	0.2 - 0.9j	0.2063 - 0.9315j	0.0321	0.0007
p_3	0.6 + 0.7j	0.6271 + 0.7372j	0.0460	0.0037
p_4	0.6 - 0.7j	0.6271 - 0.7372j	0.0460	0.0037
z_1	-0.4 + 0.8j	-0.2286 + 0.5674j	0.2889	0.0806
z_2	-0.4 - 0.8j	-0.2286 - 0.5674j	0.2889	0.0806
z_3	0.4	0.3287	0.0713	0.0000

The spectra using the true and the estimated network coefficients are plotted in Figure 16.

3. METHOD 3

a. Noise Realization 1

The coefficients converge after six iterations. Tables 33 and 34 present the results.

Table 33. ARMA(3,4) MODEL, METHOD 3, COEFFICIENTS COMPARISON, REALIZATION 1.

Coeff.	True	Estimated	Difference
a_1	-1.60	-1.5821	-0.0179
a_2	2.18	2.1352	-0.0448
a_3	-1.36	-1.3329	+0.0271
a_4	0.7225	0.7023	-0.0202
b_0	1.00	1.1733	+0.1733
b_1	0.40	0.3140	-0.0880
b_2	0.48	-0.0604	-0.5404
b_3	-0.32	-0.3978	-0.0778

Table 34. ARMA(3,4) MODEL, METHOD 3, POLE-ZERO COMPARISON, REALIZATION 1.

Poles-Zeros	True	Estimated	Distance	Radial Diff.
p_1	0.2 + 0.9j	0.1913 + 0.9j	0.0087	0.0092
p_2	0.2 - 0.9j	0.1913 - 0.9j	0.0087	0.0092
p_3	0.6 + 0.7j	0.5997 + 0.6855j	0.0145	0.0101
p_4	0.6 - 0.7j	0.5997 - 0.6855j	0.0145	0.0101
z_1	-0.4 + 0.8j	-0.4533 + 0.5691j	0.2369	0.2089
z_2	-0.4 - 0.8j	-0.4533 - 0.5691j	0.2369	0.2089
z_3	0.4	0.6406	0.2406	0.0000

b. Noise Realization 2

Using the second noise realization, the coefficients converge after four iterations. Tables 35 and 36 present the results.

Table 35. ARMA(3,4) MODEL, METHOD 3, COEFFICIENTS COMPARISON, REALIZATION 2.

Coeff.	True	Estimated	Difference
a_1	-1.60	-1.5671	+0.0329
a_2	2.18	2.2062	+0.0262
a_3	-1.36	-1.4005	-0.0405
a_4	0.7225	0.7728	+0.0503
b_0	1.00	0.9932	-0.0068
b_1	0.40	0.2049	-0.1951
b_2	0.48	0.2119	-0.2681
b_3	-0.32	-0.1654	+0.1546

Table 36. ARMA(3,4) MODEL, METHOD 3, POLE-ZERO COMPARISON, REALIZATION 2.

Poles-Zeros	True	Estimated	Distance	Radial Diff.
p_1	0.2 + 0.9j	0.1876 + 0.9336j	0.0358	0.0203
p_2	0.2 - 0.9j	0.1876 - 0.9336j	0.0358	0.0203
p_3	0.6 + 0.7j	0.5959 + 0.7051j	0.0065	0.0069
p_4	0.6 - 0.7j	0.5959 - 0.7051j	0.0065	0.0069
z_1	-0.4 + 0.8j	-0.2937 + 0.5923j	0.2333	0.0033
z_2	-0.4 - 0.8j	-0.2937 - 0.5923j	0.2333	0.0033
z_3	0.4	0.3810	0.0190	0.0000

The spectra using the true and the estimated network coefficients are plotted in Figure 17.

When using method 1 the coefficients do not converge. They oscillate about two sets of values in an alternating fashion, hence the average of the two sets is used as the estimate of the coefficients in the comparison. The parameters of the first realization converge after nine iterations using method 2, and after six iterations using method 3. The parameters of the second realization converge after four iterations using method 3. Method 3 provides the best results in terms of the coefficients and pole-zero locations for both noise realizations. The coefficients converge also earlier when using method 3 compared to the other two methods. Method 2 performs not as well but gives also reasonable results.

The spectrum due to the poles of the ARMA(3,4) models using the estimated coefficients closely resembles the original spectrum. The AR part coefficient estimates are more accurate than the MA part coefficient estimates.

D. THE ARMA(3,4) MODEL WITH OBSERVATION NOISE

Observation noise is added to the ARMA(3,4) model and resulting noisy sequence is processed via the three different methods. The observation noise is independent of the driving noise. It is white Gaussian noise with zero mean and unit variance. The signal to noise ratio is about 15 dB.

This model is illustrated in Figure 18.

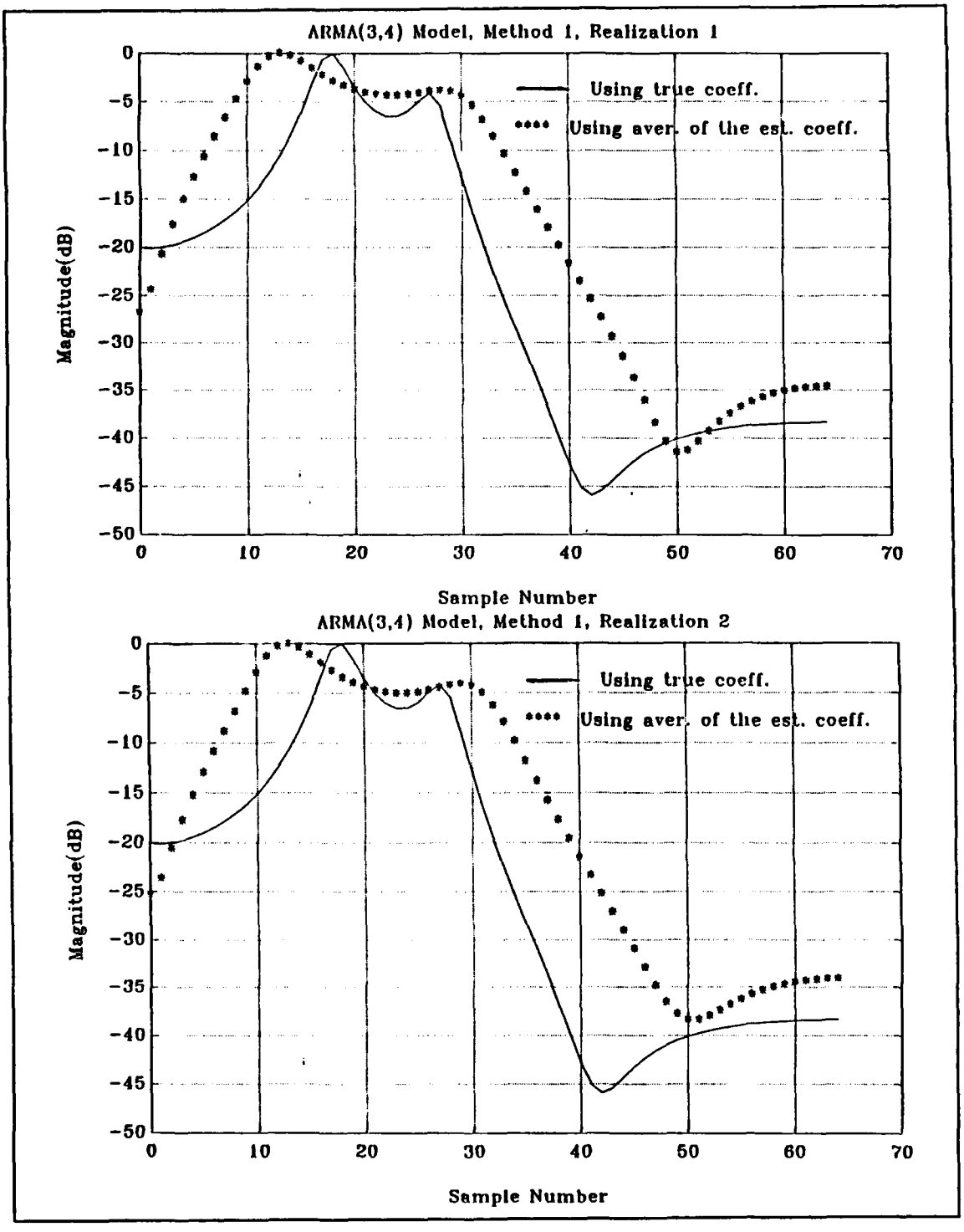

Figure 15. The Spectra of ARMA(3,4) Models, Method 1

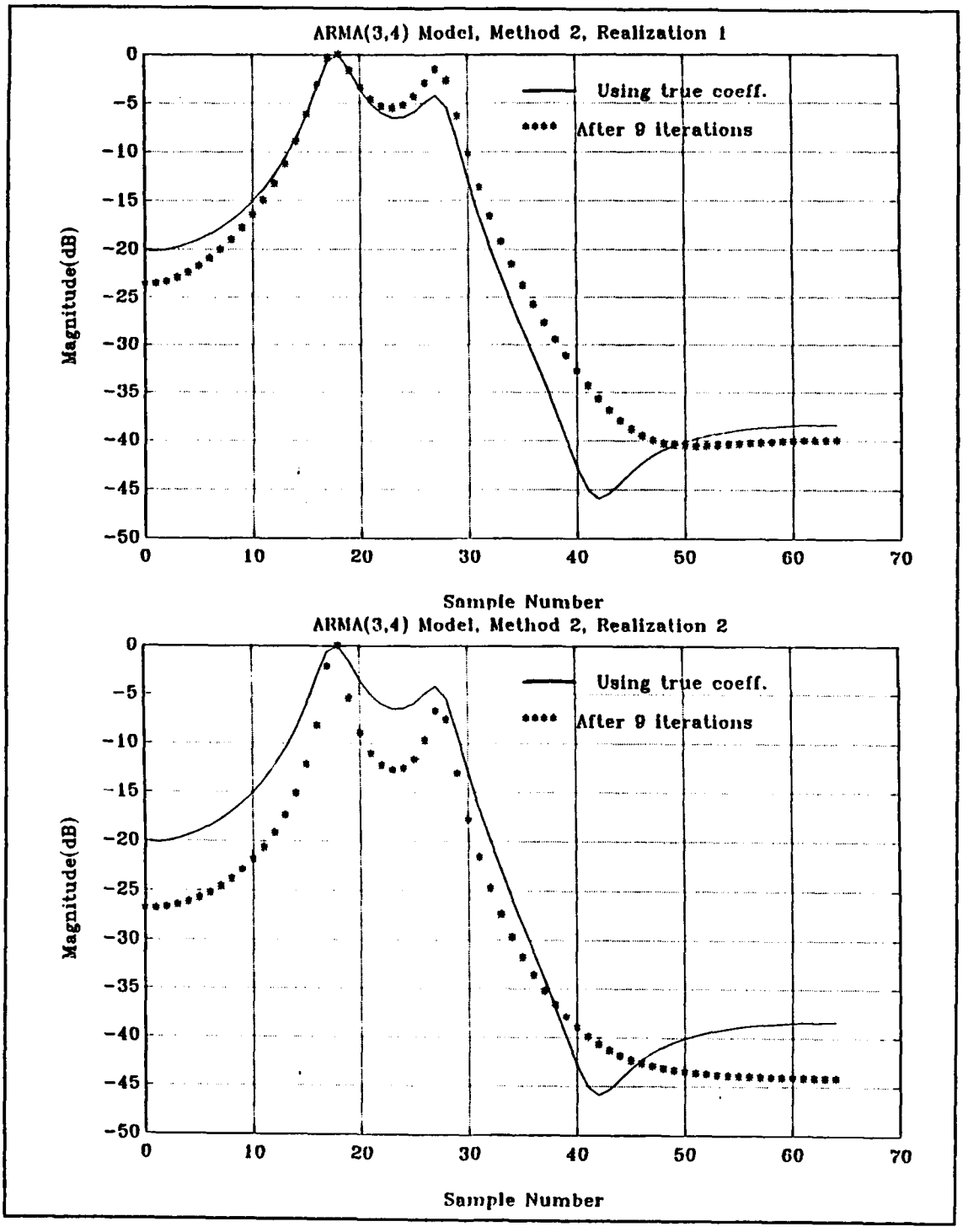

Figure 16. The Spectra of ARMA(3,4) Models, Method 2

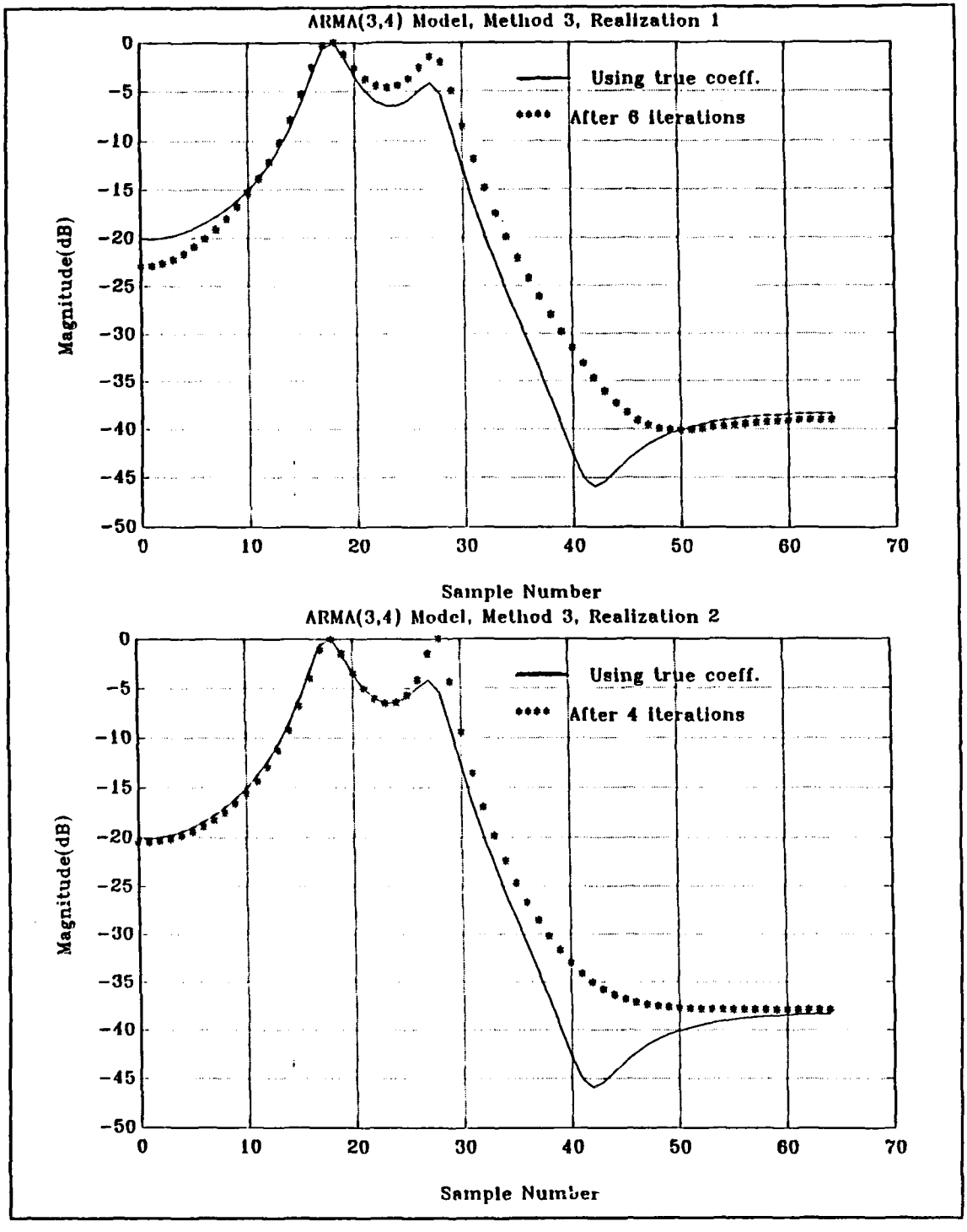

Figure 17. The Spectra of ARMA(3,4) Models, Method 3

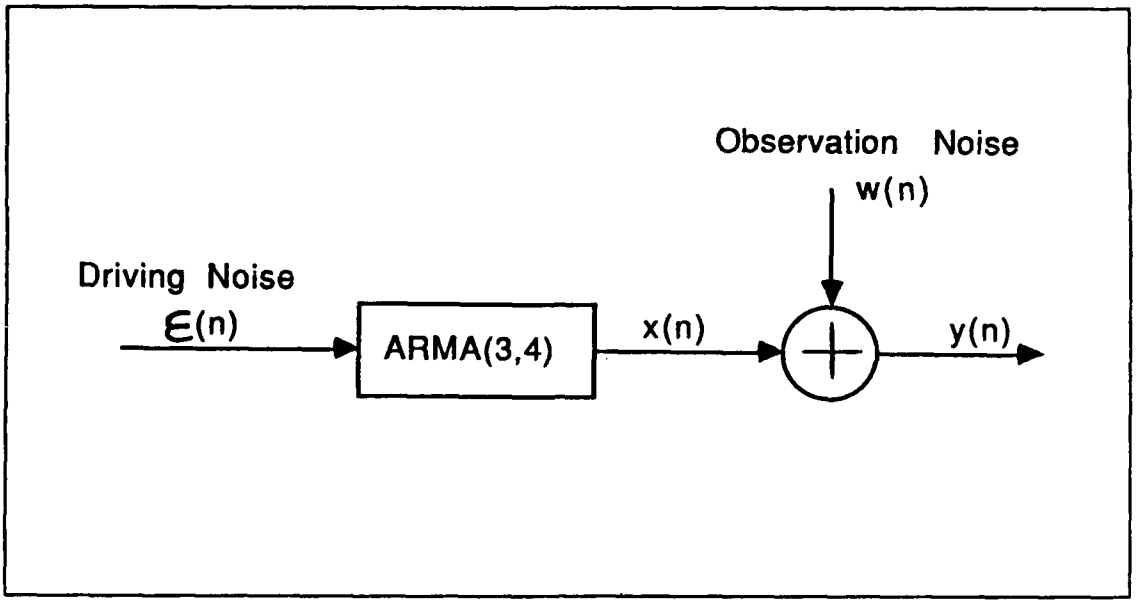

Figure 18. The ARMA(3,4) Model with Observation Noise

1. METHOD 1

The coefficients do not converge in nine iterations. Because of oscillations about two values, the average of the two values is used as an estimate of the coefficients in the comparison. Tables 37 and 38 present the results.

Table 37. ARMA(3,4) MODEL WITH OBSERVATION NOISE, METHOD 1, COEFFICIENTS COMPARISON.

Coeff.	True	Estimated	Difference
a_1	-1.60	-2.0285	-0.4285
a_2	2.18	1.9241	-0.2559
a_3	-1.36	-1.4285	-0.0685
a_4	0.7225	0.8145	+0.0920
b_0	1.00	4.5732	+3.5732
b_1	0.40	1.7315	+1.3315
b_2	0.48	-2.4570	-2.9370
b_3	-0.32	-3.3648	-3.0448

Table 38. ARMA(3,4) MODEL WITH OBSERVATION NOISE, METHOD 1, POLE-ZERO COMPARISON

Poles-Zeros	True	Estimated	Distance	Radial Diff.
p_1	0.2 + 0.9j	0.0147 + 0.8350j	0.1963	0.2010
p_2	0.2 - 0.9j	0.0147 - 0.8350j	0.1963	0.2010
p_3	0.6 + 0.7j	0.9995 + 0.4109j	0.4931	0.4721
p_4	0.6 - 0.7j	0.9995 - 0.4109j	0.4931	0.4721
z_1	-0.4 + 0.8j	-0.6723 + 0.5565j	0.3652	0.4157
z_2	-0.4 - 0.8j	-0.6723 - 0.5565j	0.3652	0.4157
z_3	0.4	0.9660	0.5660	0.0000

The spectra using the true and the estimated network coefficients are plotted in Figure 19.

2. METHOD 2

The coefficients do not converge in nine iterations. Because of oscillations about two values, the average of the two values is used as an estimate of the coefficients in the comparison. Tables 39 and 40 present the results.

Table 39. ARMA(3,4) MODEL WITH OBSERVATION NOISE, METHOD 2, COEFFICIENTS COMPARISON.

Coeff.	True	Estimated	Difference
a_1	-1.60	-2.6027	-1.0027
a_2	2.18	3.8865	+1.7065
a_3	-1.36	-2.8006	-1.4406
a_4	0.7225	1.4933	+0.7708
b_0	1.00	8.5983	+7.5983
b_1	0.40	3.4234	+3.0234
b_2	0.48	-3.2534	-3.7334
b_3	-0.32	-4.6219	-4.3019

Table 40. ARMA(3,4) MODEL WITH OBSERVATION NOISE, METHOD 2, POLE-ZERO COMPARISON

Poles-Zeros	True	Estimated	Distance	Radial Diff.
p_1	0.2 + 0.9j	0.3264 + 0.8586j	0.1330	0.1446
p_2	0.2 - 0.9j	0.3264 - 0.8586j	0.1330	0.1446
p_3	0.6 + 0.7j	0.9749 + 0.9052j	0.4273	0.1138
p_4	0.6 - 0.7j	0.9749 - 0.9052j	0.4273	0.1138
z_1	-0.4 + 0.8j	-0.6152 + 0.5170j	0.3555	0.4082
z_2	-0.4 - 0.8j	-0.6152 - 0.5170j	0.3555	0.4082
z_3	0.4	0.8323	0.4323	0.0000

The spectra using the true and the estimated network coefficients are plotted in Figure 20.

3. METHOD 3

The coefficients converge after four iterations. Tables 41 and 42 present the results.

Table 41. ARMA(3,4) MODEL WITH OBSERVATION NOISE, METHOD 3, COEFFICIENTS COMPARISON.

Coeff.	True	Estimated	Difference
a_1	-1.60	-1.4994	+0.1006
a_2	2.18	2.1311	-0.0489
a_3	-1.36	-1.3224	+0.0376
a_4	0.7225	0.7353	+0.0128
b_0	1.00	1.3898	+0.3898
b_1	0.40	0.2875	-0.1125
b_2	0.48	0.0298	-0.4502
b_3	-0.32	-0.2462	+0.0738

Table 42. ARMA(3,4) MODEL WITH OBSERVATION NOISE, METHOD 3, POLE-ZERO COMPARISON

Poles-Zeros	True	Estimated	Distance	Radial Diff.
p_1	0.2 + 0.9j	0.1823 + 0.9331j	0.0375	0.0257
p_2	0.2 - 0.9j	0.1823 - 0.9331j	0.0375	0.0257
p_3	0.6 + 0.7j	0.5674 + 0.7010j	0.0326	0.0281
p_4	0.6 - 0.7j	0.5674 - 0.7010j	0.0326	0.0281
z_1	-0.4 + 0.8j	-0.3481 + 0.4908j	0.3135	0.1532
z_2	-0.4 - 0.8j	-0.3481 - 0.4908j	0.3135	0.1532
z_3	0.4	0.4893	0.0893	0.0000

The spectra using the true and the estimated network coefficients are plotted in Figure 21.

In summary, the estimated coefficients do not converge using method 1 or 2 but they do converge after four iterations using method 3. The spectra using method 1 and 2 are poor. But the method 3 gives results close to the actual ones. The spectrum estimated with this method follows the original pattern except for the zero location. This is believed to be partially due to the imprecise MA coefficient estimation procedure.

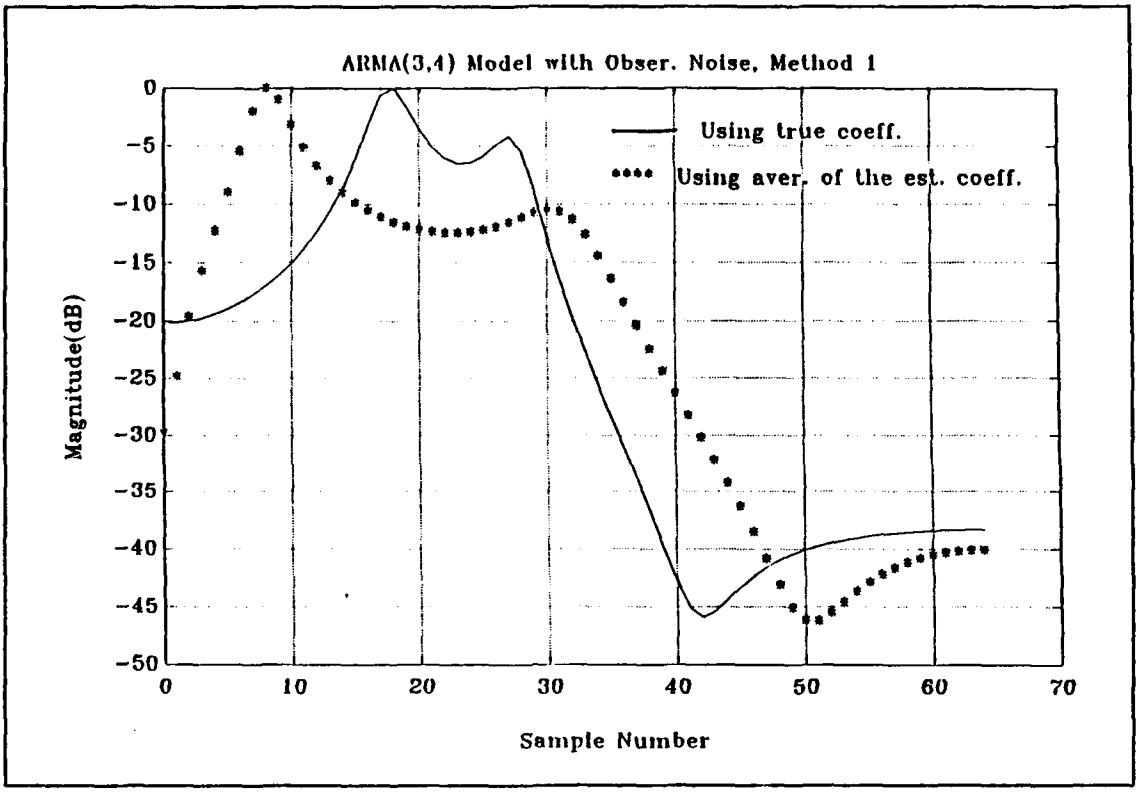

Figure 19. The Spectra of ARMA(3,4) Models with Obser. Noise, Method 1

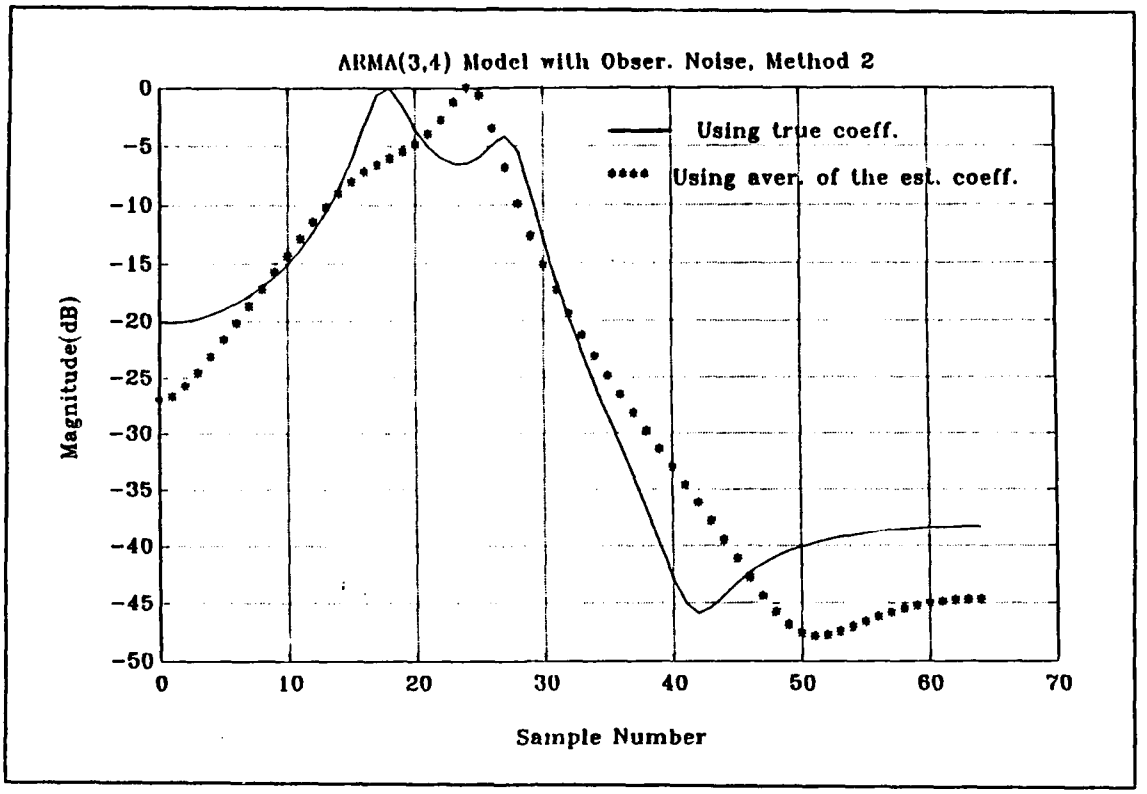

Figure 20. The Spectra of ARMA(3,4) Models with Obser. Noise, Method 2

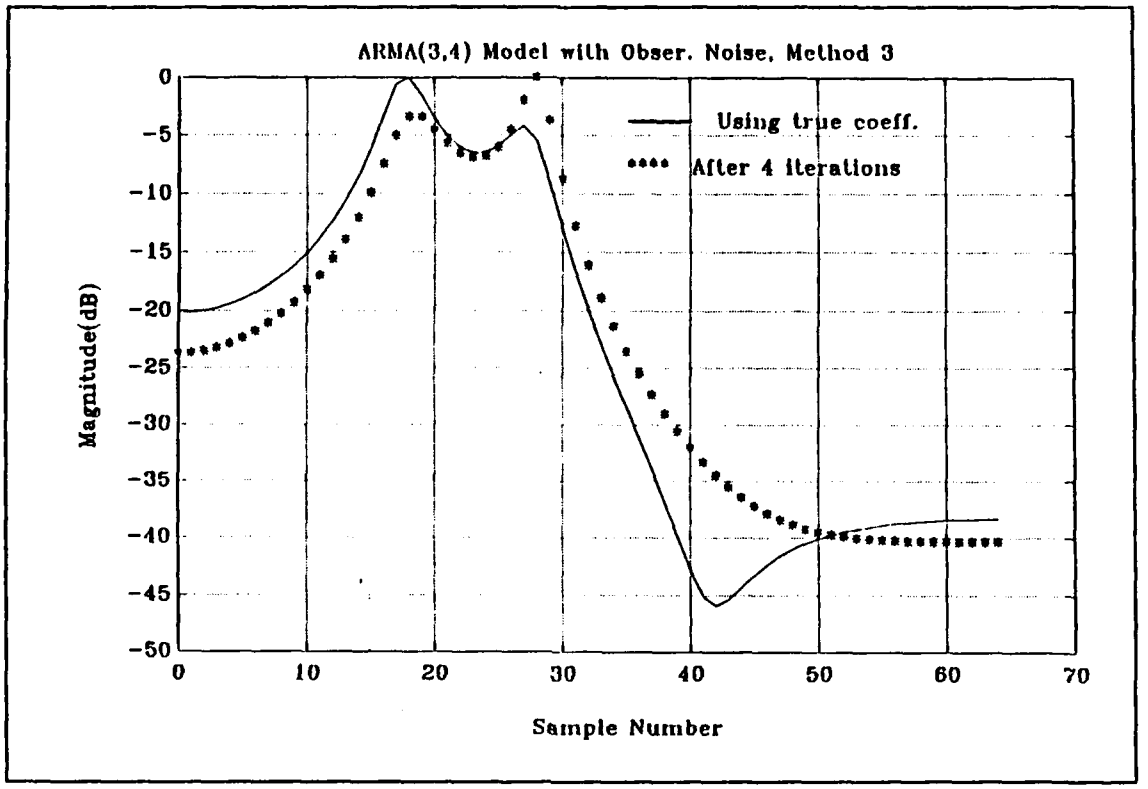

Figure 21. The Spectra of ARMA(3,4) Models with Obser. Noise, Method 3

V. CONCLUSIONS AND RECOMMENDATIONS

In this thesis, an iterative technique to estimate the coefficients of ARMA models driven by a random input is presented. The AR parameters are estimated initially, then MA parameters are estimated assuming the AR parameters have been compensated for. In an iterative fashion MA and AR contributions are removed from the original data allowing improved AR and MA coefficient estimates. Three different AR estimation methods are experimentally explored. The third method provides the best result in terms of the true and estimated coefficients and pole-zero locations. This method uses the pseudoinverse with correlation function values starting at the zero lag after removing the MA influence. For models of real time data which have an odd number of poles, one should address the issue of the DC component (i.e., remove it). Simulation results for an odd ordered pole model is presented in Appendix C.

Also, by examining the spectra of the models, we can say estimation of poles is obtained more accurately than the estimation of zeros. The Cholesky factorization is used for the estimation of the MA part coefficients but it is an approximate solution. For that reason, we do not expect superior results for the MA estimation part.

Further research should concentrate on improving the MA part coefficients estimation.

APPENDIX A. COMPUTER PROGRAM TO ESTIMATE COEFFICIENTS

```
%**************************************
%  This program written by          *
%  Gurhan Kayahan for IBM PC/AT     *
%  Matlab package.                  *
%**************************************
%**************************************
% GENERATION OF ARMA TIME SERIES    *
%**************************************
i=sqrt(-1);
roots1=[ -0.25 + 0.5i  - 0.25 - 0.5i]; change zeros for each model
roots2= [0.6 + 0.4i  0.6 - 0.4i  0.8]; change poles for each model
b=poly(roots1); find true coefficients of MA part
a=poly(roots2); find true coefficients of AR  part
a=real(a);
t=1:1000;
y=t*0;
rand('normal'); generate random signal
stddev=sqrt(1);
rnum=stdev*rand(t);
y1=filter(b,a,rnum); filter random signal through ARMA model
x= [ones(1,999)]; generate step signal
y2=filter(b,a,x); find step response of ARMA model to discard transient response
y=y1(52:252);
%*************************************************************************
% CALCULATE AR COEFFICIENTS USING YULE-WALKER APPROACH  *
%*************************************************************************
r1=xcorr(y);
r1=r1(201:399);
r=flipy(r1);
row=r(194:196);
col=flipy(r(192:194));
a=toeplitz(col,row);
b=flipy(-r(191:193));
c=b';
x=b*inv(a); square matrix inverse
a1=x';
%*********************************************************************
% CALCULATE MA COEFFICIENTS USING CHOLESKY FACTORIZATION  *
%*********************************************************************
d= [1];
n= [1 a1(1) a1(2) a1(3)];
z=filter(n,d,y);
z1=z(4:201);
rz1=xcorr(z1);
rz=rz1(198:395);
i=1;
while i<199
  rz3(i)=rz(i)/198;
```

```
  i=i+1;
end
rz2=rz3(1:3);
rzt=toeplitz(rz2);
cho=chol(rzt);
b1=cho(1,:);
%*************************************************
% ROUTINE FOR ESTIMATION OF AR COEFFICIENTS  *
%*************************************************
i=2;
while i<11
 d= [b1(i-1,1) b1(i-1,2) b1(i-1,3)];
 n= [1];
 zar=filter(n,d,y);
 rar=xcorr(zar);
 r1=rar(201:399);
 r=flipy(r1);
 row=r(194:196);change correlation lags for each method
 col=flipy(r(192:194));change correlation lags for each method
 g=toeplitz(col,row);
 e=flipy(-r(191:193));change correlation lags for each model
 f=e';
 x=f*inv(g);square matrix inverse (replace with x=pinv(g) for pseudo inverse)
 a(i,:)=x';
 a1= [a1', a(i,:)'];
%*************************************************
% ROUTINE FOR ESTIMATION OF MA COEFFICIENTS   *
%*************************************************
d= [1];
n= [1 a(i,1) a(i,2) a(i,3)];
z=filter(n,d,y);
z1=z(4:201);
rz1=xcorr(z1);
rz=rz1(198:395);
k=1;
while k<199
  rz3(k)=rz(k)/198;
  k=k+1;
end
rz2=rz3(1:3);
rzt=toeplitz(rz2);
cho=chol(rzt);
bt(i,:)=cho(1,:);
b1= [b1', bt(i,:)'];
  e(i-1)=sum( (a1(i,:)-a1(i-1,:)). ^ 2 )/3+sum( (b1(i,:)-b1(i-1,:)). ^ 2 )/3;
if e(i-1) < 0.0001
    a1=a1;
    b1=b1;
else
    i=i+1;
end
end
```

APPENDIX B. COMPUTER PROGRAM TO CALCULATE TRUE AND ESTIMATED SPECTRA

```
%****************************************
% CALCULATION OF TRUE SPECTRUM
%****************************************
a= [1  -1.20  0.52]; change true poles for each models
b= [1  0.50  0.3125]; change true zeros for each models
a1= [a zeros(1,123)]; zero padding
b1= [b zeros(1,124)]; zero padding
fa1=fft(a1); fast fourier transform of AR part
fb1=fft(b1); fast fourier transform of MA part
mfa1=abs(fa1). ^ 2;
mfb1=abs(fb1). ^ 2;
c=mfb1./mfa1;
a=max(c);
i=1;
while i<129
  c(i)=c(i)/a;
  i=i+1;
end
spec=10*log10(c);
%****************************************
% CALCULATION OF ESTIMATED SPECTRUM
%****************************************
a= [1  -1.1531  0.5013]; change estimated poles for each models
b= [1.0174  0.4732  0.1202]; change estimated zeros for each models
a1= [a zeros(1,123)];
b1= [b zeros(1,124)];
fa1=fft(a1);
fb1=fft(b1);
mfa1=abs(fa1). ^ 2;
mfb1=abs(fb1). ^ 2;
c=mfb1./mfa1;
a=max(c);
i=1;
while i<129
  c(i)=c(i)/a;
  i=i+1;
end
spec7=10*log10(c);
%****************************************************
% PLOTTING OF TRUE AND ESTIMATED SPECTRUM
%****************************************************
t=0:127;
t=t';
plot(t(1:65),spec7(1:65),'*',t(1:65),spec(1:65))
title('ARMA(2,2) Model, Method 3, Realization 1')
xlabel('Sample Number')
ylabel('Magnitude(dB)')
grid
text(25,-5,'----  Using true coefficients')
```

```
text(25,-9,'****  After 4 iterations')
delete spec1.met
meta spec1
```

APPENDIX C. SIMULATION RESULTS FOR AN ARMA(2,3) MODEL

For models with an odd number of poles, one of the poles must be located on the real axis in the z-domain to generate real time data.

The pole-zero locations for this model are illustrated in Figure 22.

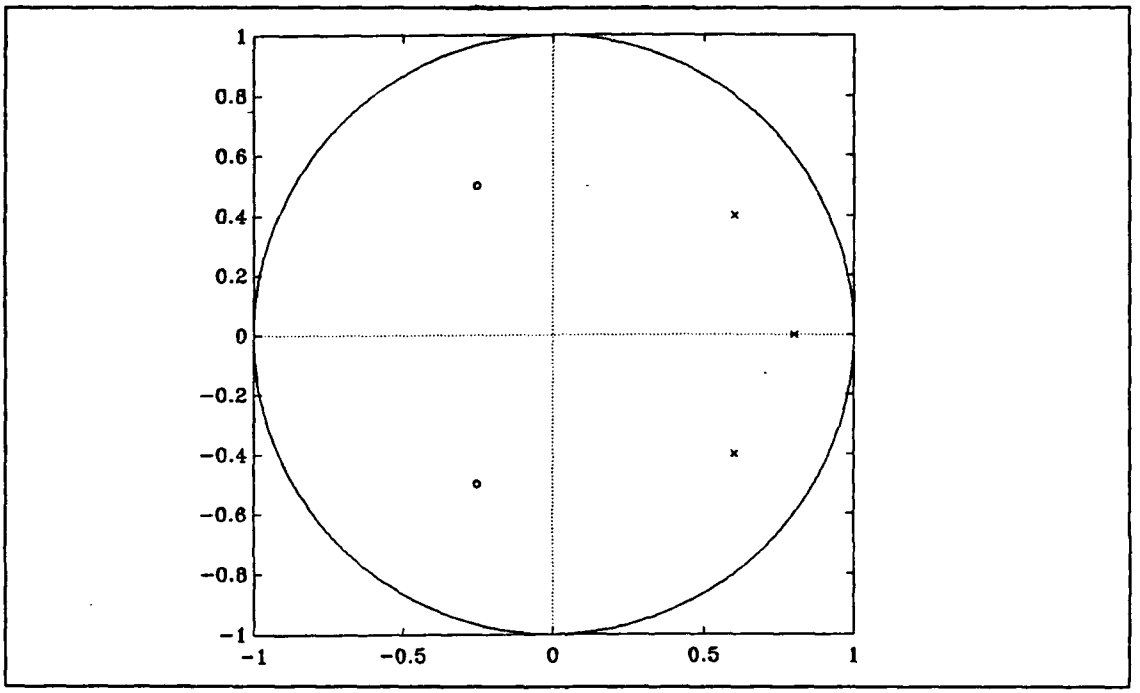

Figure 22. The ARMA(2,3) Model Pole-zero Locations

1. METHOD 1

a. Noise Realization 1

The coefficients do not converge in nine iterations. Because of oscillations about two values, the average of the two values is used as an estimate of the coefficients in the comparison. Tables 43 and 44 present the results.

Table 43. ARMA(2,3) MODEL, METHOD 1, COEFFICIENTS COMPARISON, REALIZATION 1.

Coeff.	True	Estimated	Difference
a_1	-2.00	-1.9425	+0.0575
a_2	1.48	1.4028	-0.0772
a_3	-0.416	-0.3639	+0.0521
b_0	1.00	1.0562	+0.0562
b_1	0.50	0.5184	-0.0184
b_2	0.3125	0.1681	-0.1444

Table 44. ARMA(2,3) MODEL, METHOD 1, POLE-ZERO COMPARISON, REALIZATION 1.

Poles-Zeros	True	Estimated	Distance	Radial Diff.
p_1	0.6 + 0.4j	0.6526 + 0.3809j	0.0559	0.0596
p_2	0.6 - 0.4j	0.6526 - 0.3809j	0.0559	0.0596
p_3	0.8	0.6373	0.1627	0.0000
z_1	-0.25 + 0.5j	-0.2454 + 0.3145j	0.1855	0.1989
z_2	-0.25 - 0.5j	-0.2454 - 0.3145j	0.1855	0.1989

b. Noise Realization 2

Using a different noise realization, the coefficients converge after seven iterations. Tables 45 and 46 present the results.

Table 45. ARMA(2,3) MODEL, METHOD 1, COEFFICIENTS COMPARISON, REALIZATION 2.

Coeff.	True	Estimated	Difference
a_1	-2.00	-2.8571	-0.8571
a_2	1.48	2.7525	1.2725
a_3	-0.416	-0.9162	+1.3322
b_0	1.00	1.3418	+0.3418
b_1	0.50	0.1291	-0.3709
b_2	0.3125	-0.4401	-0.7526

Table 46. ARMA(2,3) MODEL, METHOD 1, POLE-ZERO COMPARISON, REALIZATION 2.

Poles-Zeros	True	Estimated	Distance	Radial Diff.
p_1	0.6 + 0.4j	0.8301 + 0.2765j	0.2611	0.2664
p_2	0.6 - 0.4j	0.8301 - 0.2765j	0.2611	0.2664
p_3	0.8	1.1968	0.3968	0.0000
z_1	-0.25 + 0.5j	-0.6229	0.6237	1.1071
z_2	-0.25 - 0.5j	-0.5266	0.9236	2.0344

2. METHOD 2

a. Noise Realization 1

The coefficients do not converge in nine iterations. Because of oscillations about two values, the average of the two values is used as an estimate of the coefficients in the comparison. Tables 47 and 48 present the results.

Table 47. ARMA(2,3) MODEL, METHOD 2, COEFFICIENTS COMPARISON, REALIZATION 1.

Coeff.	True	Estimated	Difference
a_1	-2.00	-1.6534	+0.3466
a_2	1.48	0.9727	-0.5073
a_3	-0.416	-0.2044	+0.2116
b_0	1.00	1.2677	+0.2677
b_1	0.50	0.8274	+0.3274
b_2	0.3125	0.3741	+0.0616

Table 48. ARMA(2,3) MODEL, METHOD 2, POLE-ZERO COMPARISON, REALIZATION 1.

Poles-Zeros	True	Estimated	Distance	Radial Diff.
p_1	0.6 + 0.4j	0.5267 + 0.2515j	0.1656	0.1425
p_2	0.6 - 0.4j	0.5267 - 0.2515j	0.1656	0.1425
p_3	0.8	0.6001	0.1999	0.0000
z_1	-0.25 + 0.5j	-0.3263 + 0.4349j	0.1006	0.1807
z_2	-0.25 - 0.5j	-0.3263 - 0.4349j	0.1006	0.1807

b. Noise Realization 2

Using a second noise realization, the coefficients do not converge. Because of oscillations about two values, the average of the two values is used as an estimate of the coefficients in the comparison. Tables 49 and 50 present the results.

Table 49. ARMA(2,3) MODEL, METHOD 2, COEFFICIENTS COMPARISON, REALIZATION 2.

Coeff.	True	Estimated	Difference
a_1	-2.00	-1.1005	+0.8995
a_2	1.48	-0.1144	-1.5944
a_3	-0.416	0.3945	+0.8105
b_0	1.00	1.8387	+0.8387
b_1	0.50	1.3064	+0.8064
b_2	0.3125	0.4440	+0.1315

Table 50. ARMA(2,3) MODEL, METHOD 2, POLE-ZERO COMPARISON, REALIZATION 2.

Poles-Zeros	True	Estimated	Distance	Radial Diff.
p_1	0.6 + 0.4j	0.8145 + 0.2883j	0.2418	0.2478
p_2	0.6 - 0.4j	0.8145 - 0.2883j	0.2418	0.2478
p_3	0.8	-0.5285	1.3285	3.1415
z_1	-0.25 + 0.5j	-0.3553 + 0.3395j	0.1919	0.3444
z_2	-0.25 - 0.5j	-0.3553 - 0.3395j	0.1919	0.3444

3. METHOD 3

a. Noise Realization 1

The coefficients converge after five iterations. Tables 51 and 52 present the results.

Table 51. ARMA(2,3) MODEL, METHOD 3, COEFFICIENTS COMPARISON, REALIZATION 1.

Coeff.	True	Estimated	Difference
a_1	-2.00	-1.1057	+0.8943
a_2	1.48	0.2111	-1.2689
a_3	-0.416	0.0890	+0.5050
b_0	1.00	1.8594	+0.8594
b_1	0.50	1.4885	-0.9885
b_2	0.3125	0.8320	-0.5195

Table 52. ARMA(2,3) MODEL, METHOD 3, POLE-ZERO COMPARISON, REALIZATION 1.

Poles-Zeros	True	Estimated	Distance	Radial Diff.
p_1	0.6 + 0.4j	0.6493 + 0.2j	0.2059	0.2892
p_2	0.6 - 0.4j	0.6493 - 0.2j	0.2059	0.2892
p_3	0.8	-0.1928	0.9928	3.1415
z_1	-0.25 + 0.5j	-0.4002 + 0.5360j	0.0509	0.1777
z_2	-0.25 - 0.5j	-0.4002 - 0.5360j	0.0509	0.1777

b. Noise Realization 2

Using a second noise realization, the coefficients do not converge. Because of oscillations about two values, the average of the two values is used as an estimate of the coefficients in the comparison. Tables 53 and 54 present the results.

Table 53. ARMA(2,3) MODEL, METHOD 3, COEFFICIENTS COMPARISON, REALIZATION 2.

Coeff.	True	Estimated	Difference
a_1	-2.00	-1.7973	+0.2027
a_2	1.48	0.9959	-0.4841
a_3	-0.416	-0.9354	-0.5194
b_0	1.00	1.2540	+0.2540
b_1	0.50	0.6150	+0.1150
b_2	0.3125	0.0151	-0.2974

Table 54. ARMA(2,3) MODEL, METHOD 3, POLE-ZERO COMPARISON, REALIZATION 2.

Poles-Zeros	True	Estimated	Distance	Radial Diff.
p_1	0.6 + 0.4j	0.1263 + 0.7679j	0.5997	0.2564
p_2	0.6 - 0.4j	0.1263 - 0.7679j	0.5997	0.2564
p_3	0.8	1.5446	0.7446	0.0000
z_1	-0.25 + 0.5j	-0.4645	0.5440	1.1071
z_2	-0.25 - 0.5j	-0.0259	0.5479	1.1071

Simulation results have shown that for the ARMA(2,3) model with one pole on positive real axis, realization 2 gives poor results. This is because the DC component for this realization is - 0.9017 while it is -0.0580 for realization 1. For parameter estimations the DC component should be removed. Therefore the data is filtered effectively removing the pole on the positive real axis (i.e., DC component is removed and the ARMA(2,3) model becomes ARMA(2,2) model). The results of this ARMA(2,2) model was presented in chapter IV-B. The ARMA(2,3) and ARMA(2,2) true and estimated spectra using method 1 are presented in Figure 23. Method 2 results are presented in Figure 24 and method 3 results are presented in Figure 25. The simulation shows that the method 3 gives good results as long as the DC component is small.

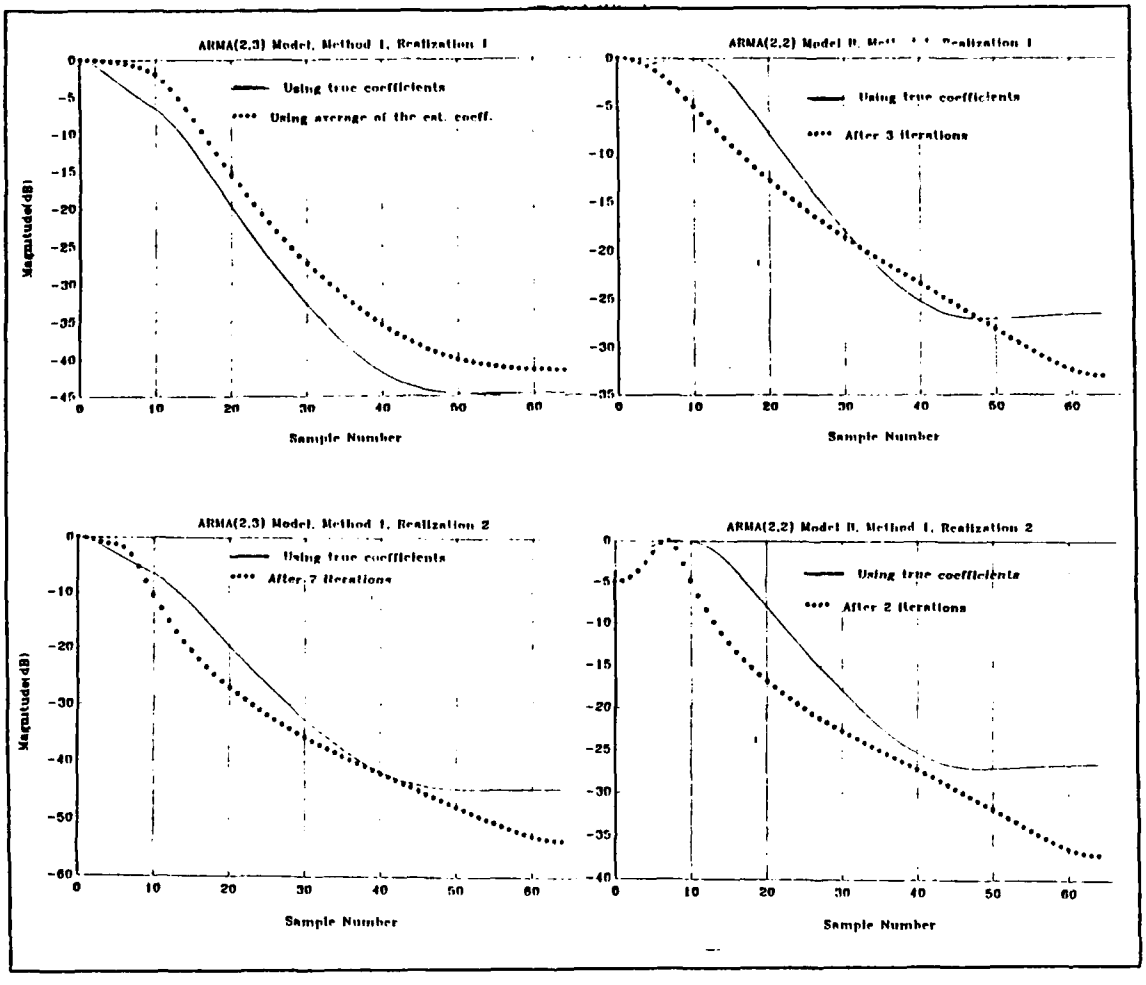

Figure 23. The Spectra of ARMA(2,3) and ARMA(2,2) Model-B, Method 1

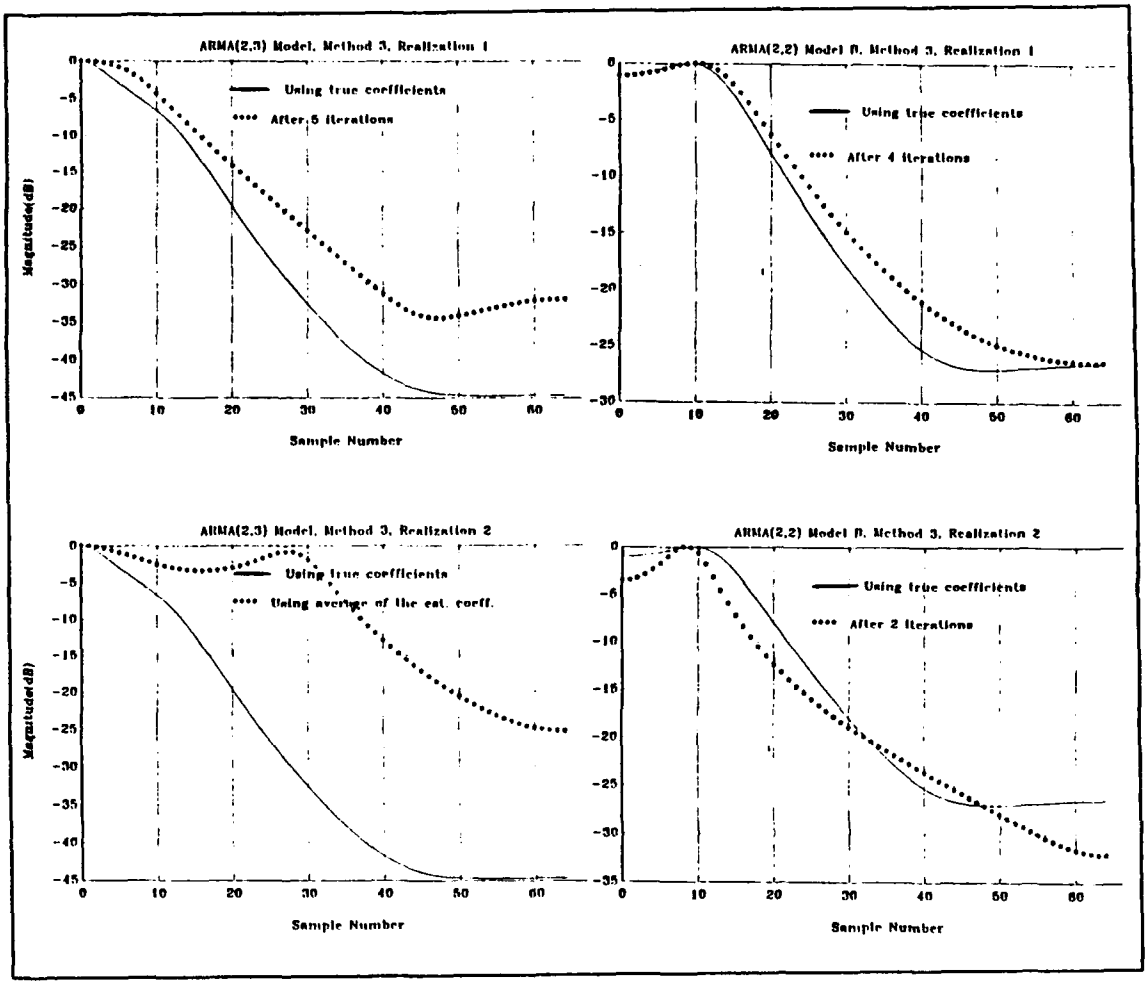

Figure 24. The Spectra of ARMA(2,3) and ARMA(2,2) Model-B, Method 2

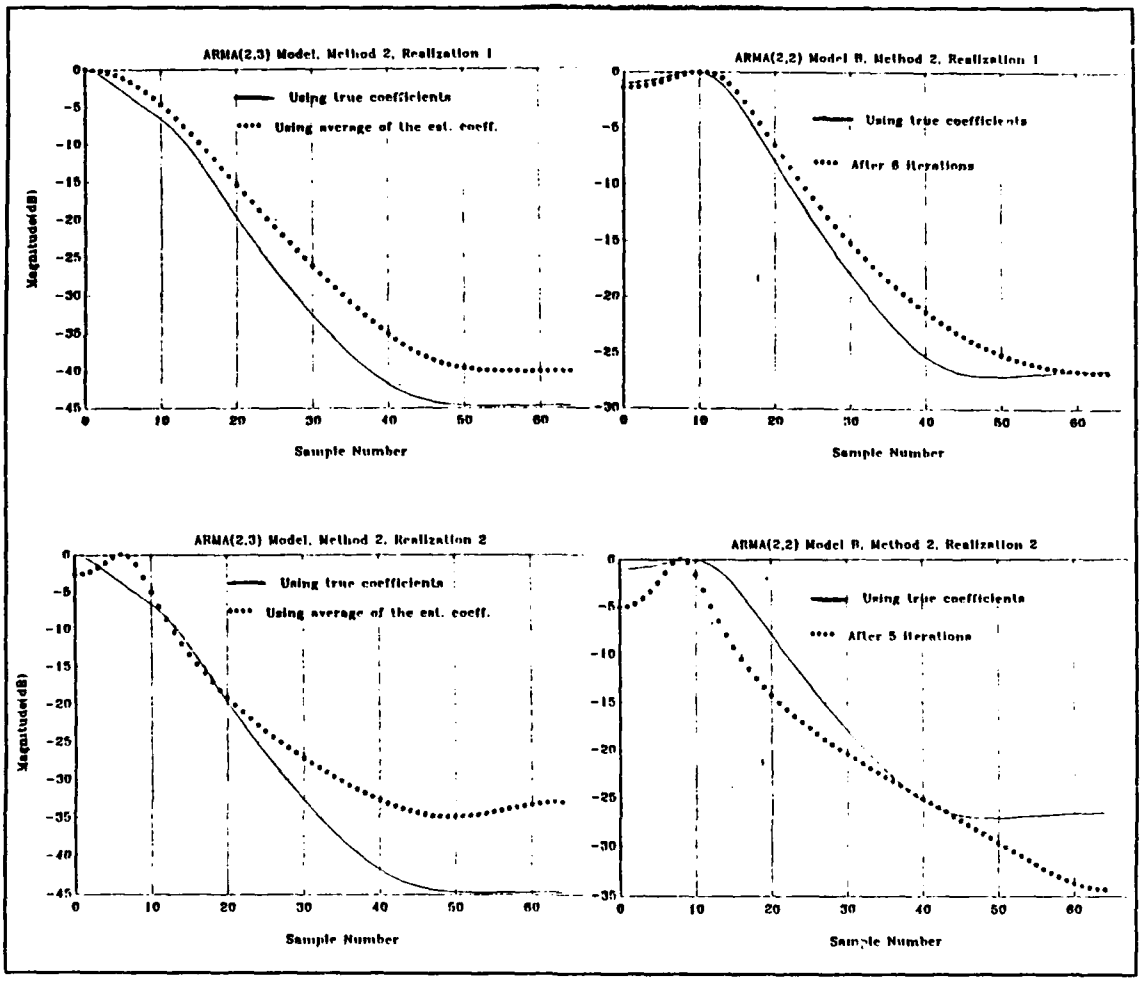

Figure 25. The Spectra of ARMA(2,3) and ARMA(2,2) Model-B, Method 3

LIST OF REFERENCES

1. Priestley, M. B., *Spectral Analysis and Time Series*, Vol. 1, Academic Press, 1981.

2. Marple, S. Lawrence, Jr., *Digital Spectral Analysis with Applications*, pp. 174-185, Prentice-Hall, Inc., 1987.

3. Wold, H., *A Study in the Analysis of Stationary Time Series*, Almqvist & Wiksell, Stockholm, 1954.

4. Kolmogorov, A. N., *Interpolation und Extrapolation von Stationaren Zufalligen Folgen*, Bull. Acad. Sci. USSR Ser. Math, Vol. 5, pp. 3-14, 1941.

5. Kay, Steven M., *Modern Spectral Estimation*, pp. 222-224, Prentice-Hall, New Jersey, 1988.

6. Strum, Robert D., and Kirk, Donald E., *Discrete Systems and Digital Signal Processing*, pp. 352-354, Addison-Wesley Publishing Company, 1988.

7. Orfanidis, Sophocles J., *Optimum Signal Processing*, pp. 185-189, Macmillan Publishing Company, 1985.

8. Box, G.E.P., and Jenkins, G.M., *Time Series Analysis, Forecasting and Control*, Holden-Day, Inc., San Fransisco, 1970.

9. Whittle, P., *Prediction and Regulation*, New York, Van Nostrand Reinhold, 1963.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, CA 93943-5002	2
3.	Chairman, Code 62 Naval Postgraduate School Monterey, CA 93943-5000	1
4.	Professor Ralph Hippenstiel, Code 62Hi Naval Postgraduate School Monterey, CA 93943-5000	1
5.	Professor Murali Tummala, Code 62Tu Naval Postgraduate School Monterey, CA 93943-5000	1
6.	Gurhan Kayahan Feneryolu Cad. Kilic Sok. No:20 10 Tatli Huzur Apt. Kalamis - Istanbul, TURKEY	2
7.	Deniz Kuvvetleri Komutanligi Personel Egitim Daire Baskanligi Bakanliklar - Ankara, TURKEY	3
8.	Deniz Harp Okulu Ogretim Kurulu Baskanligi Tuzla - Istanbul, TURKEY	2
9.	O. D. T. U. Kutuphanesi Orta Dogu Teknik Universitesi Ankara, TURKEY	1
10.	Naval Ocean Systems Center Attn. Dr. C. E. Persons, Code 732 San Diego, CA 92152	1